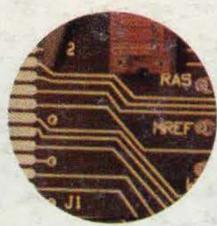
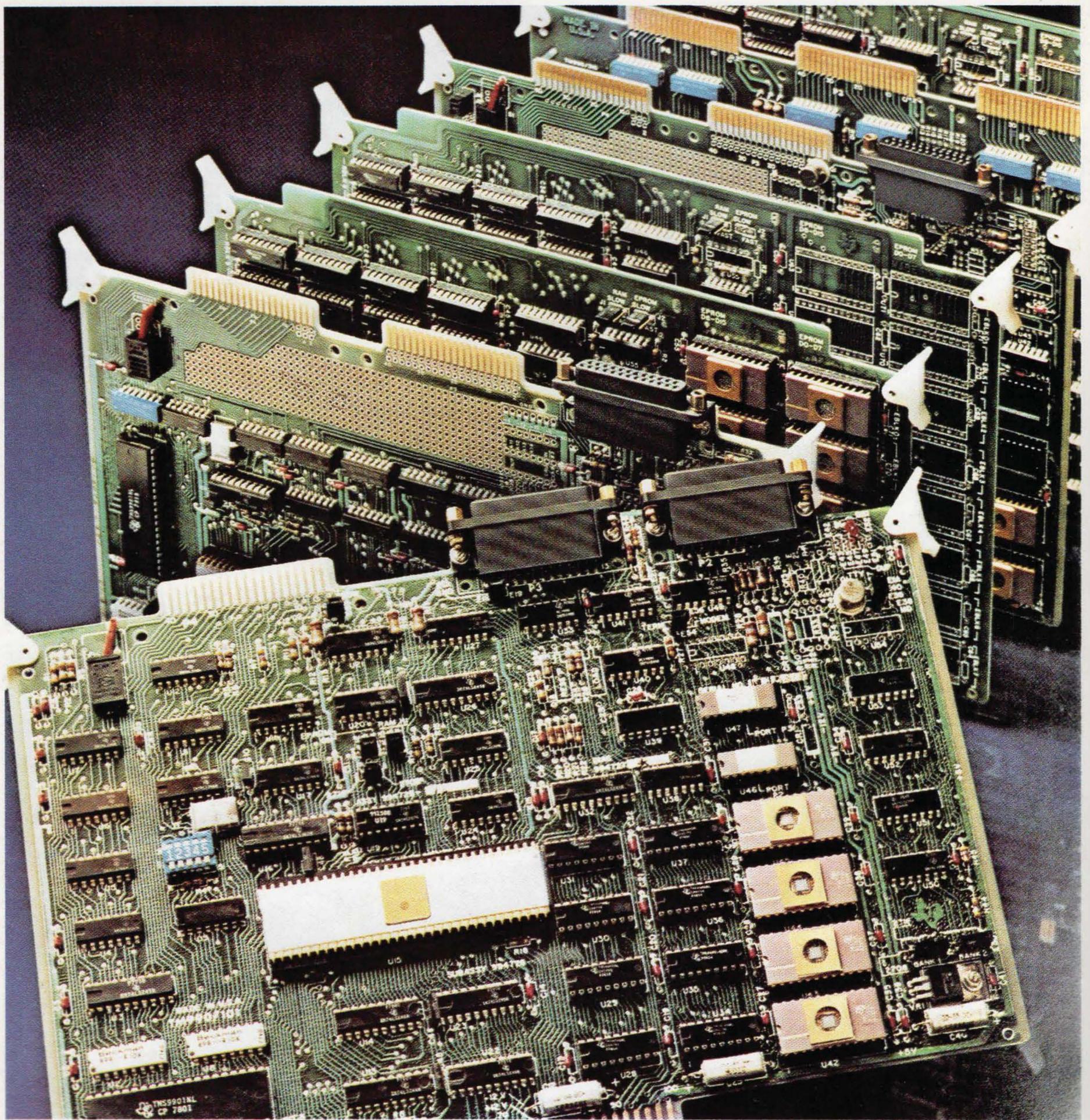
UNA PUBBLICAZIONE
DEL GRUPPO EDITORIALE JACKSON

ANNO 2 N. 2

Bit

MICROPROCESSORS-HARDWARE - SOFTWARE
HOME & PERSONAL COMPUTERS

L. 1500

Jackson

La Texas Instruments annuncia il primo modulo di microcomputer a 16 BIT con linguaggio BASIC. Più facile da programmare, ampliato nella capacità di memorizzazione e di comunicazione

Sono disponibili presso i distributori autorizzati T.I., dei veri e propri computers su singola scheda.
Stiamo parlando dei nuovi moduli di microcomputers TM 990/101M.
La maggiore estensione di memoria, la possibilità di comunicazioni simultanee su due porte (con protocolli di comunicazione già implementati su board), e la disponibilità del POWER BASIC, linguaggio ad alto livello che rende estremamente veloce la programmazione, sono la caratteristiche fondamentali di questi moduli.

Questi nuovi moduli della Texas Instruments riducono i tempi di progetto e di sviluppo, il numero di

componenti del sistema ed i costi di realizzazione.

Queste unità sono preassemblate, pretestate e pronte all'uso, e questo garantisce un'alta affidabilità al sistema.

Estensione di memoria quattro volte più grande

I microcomputers TM 990/101M possiedono una capacità di memoria RAM statica su board che va da 1K a 2K parole da 16 bits.
L'estensione di EPROM va da 2K a 4K parole da 16 bits.

Capacità di comunicazione doppia

Sul board sono disponibili due porte per comunicazioni seriali: una usata per utilizzazioni "remote" (terminali o modem), l'altra per utilizzazioni "locali" (terminali EIA, teletype, microterminale TM 990/301 della T.I.).

Il meglio nell'ambito dei microcomputers

Nella serie TM 990 della T.I. troverete la più ampia scelta possibile per prestazioni/costo, di microcomputers a 16 bits che possono soddisfare le vostre necessità a livello di sistema.
Ideali per valutazioni su microprocessori e per accelerare il progetto di sistemi a microprocessori, rappresentano una valida alternativa in produzione.
Caratteristica fondamentale di tutti i moduli è la completa compatibilità a livello di set di istruzioni con gli altri membri dell'intera famiglia TM 990/9900.

Per valutazioni ed applicazioni OEM

- TM 990/100M - Utilizza il microprocessore TMS 9900 a 16 bit in tecnologia N MOS, della Texas Instruments.

  1K bytes di RAM statica, 2K bytes di EPROM, e porte di I/O seriali e parallele programmabili, fanno di questa unità un potente microcomputer su singola scheda.

- TM 990/180M - Utilizza il microprocessore TMS 9980 a 16 bit, in tecnologia N MOS, della T.I., che consente di lavorare a 2,5 MHz, e che contengono una interfaccia a 8 bit per la memoria, minimizza i costi dei sistemi più piccoli.

Per espansioni di memoria

- TM 990/201 - 8K bytes di EPROM e 4K bytes di RAM statica. Espandibile a 32K bytes di EPROM e 16K bytes di RAM.
- TM 990/206 - 8K bytes di RAM statica espandibile a 16K bytes.

Per ingresso dati e monitoraggio

- TM 990/301 - Consente l'ingresso di dati di programma, la visualizzazione e la modifica del contenuto dei registri interni e della memoria sotto controllo software (TIBUG).

Per espandere le linee di I/O

- TM 990/310 - Modulo di espansione delle linee di I/O a 48 bit.

Per sviluppare software e per la produzione

- TM 990/401 - Monitor interattivo di debug (TIBUG) già programmato in EPROM.
- TM 990/402 - Assemblatore line-by-line già programmato in EPROM.
- TM 990/450 - POWER BASIC già programmato in 8K bytes di E-PROM.
- TM 990/451 - POWER BASIC già programmato in 12K bytes di E-PROM.

Sono inoltre disponibili una vasta gamma di connettori e cavi per gli usi più comuni, schede di prototipizzazione, ed uno chassis a 4-slot (TM 990/510) già predisposto per ospitare 4 schede della famiglia TM 990.

Supporto software

I moduli di microcomputer della serie TM 990 sono completamente supportati dal sistema di sviluppo AMPL (Advanced Microprocessor Prototyping Laboratory) della Texas Instruments.

L'AMPL consente di effettuare trace a 10 MHz ed emulazioni per i microprocessori TMS 9900, 9980, SBP 9900, TMS 9940 e per tutti i microprocessori della T.I. che saranno prodotti in futuro.

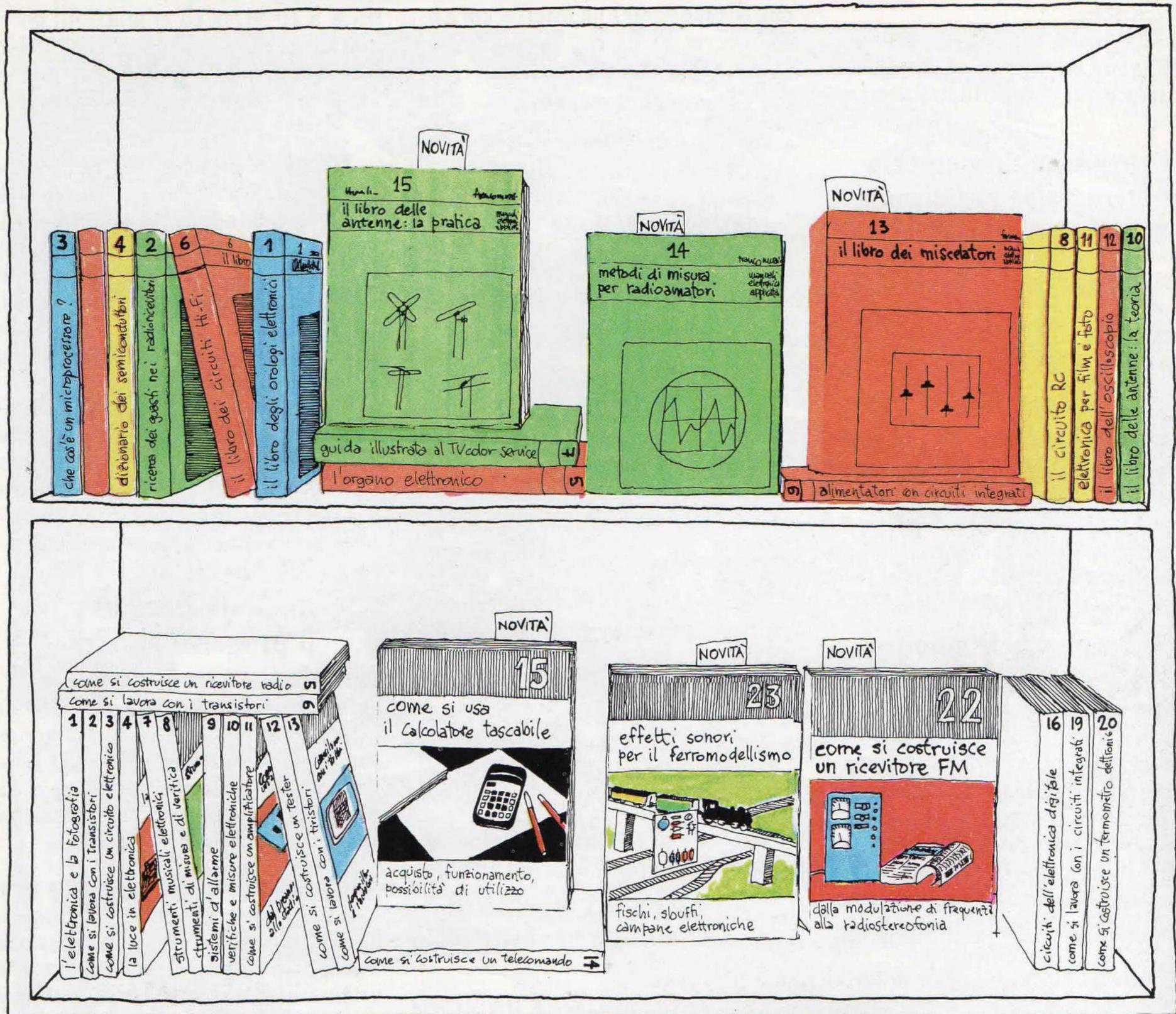
L'AMPL è disponibile nella configurazione con sistema a floppy disk o in quella con sistema a dischi rigidi in modo da soddisfare le più svariate esigenze degli utenti. I programmi possono essere editati, assemblati, lincati, caricati ed eseguiti più velocemente che con i sistemi tradizionali a nastri o a cassette.

L'emulazione del TMS 9900/9980 consente di sviluppare e di debuggare il software direttamente su un modulo TM 990 visualizzando e controllando, contemporaneamente, le operazioni per mezzo del sistema di sviluppo AMPL.

Per salvaguardare il progetto di oggi dalle innovazioni di domani

I microcomputers della serie TM 990 e l'AMPL si integrano perfettamente con l'intera famiglia 990/9900 rendendola veramente completa e di estema potenzialità nell'ambito delle applicazioni a 16 bit.

Questa famiglia è composta da una vasta gamma di microprocessori, microcomputers e minicomputers che usano tutti la stessa architettura orientata verso la memoria, lo stesso set di istruzioni; sono tutti completamente compatibili e sono previsti per essere supportati dallo stesso sistema di sviluppo.

Con la famiglia 990/9900 il progettista può scegliere la soluzione a costo più basso, e più soddisfacente alle sue esigenze, senza preoccuparsi degli sviluppi futuri del suo prodotto, in quanto la completa compatibilità software rende il progetto software direttamente utilizzabile su prodotti che si possono diversificare per sofisticazione hardware.
Per maggiori informazioni contattate i nostri distributori autorizzati oppure l'ufficio Promozione Commerciale, Cittaducale - RIETI

manuali di elettronica applicata

- ☐ 1 Pelka - Il libro degli orologi elettronici, L. 4.400
- ☐ 2 Renardy/Lummer - Ricerca dei guasti nei radioricevitori, 2ª edizione 1978, L. 4000
- ☐ 3 Pelka - Cos'è un microprocessore? 2ª edizione 1978, L. 4000
- ☐ 4 Buscher/Wiegelmann - Dizionario dei semiconduttori, L. 4.400
- ☐ 5 Böhm - L'organo elettronico, L. 4.400
- ☐ 6 Kühne/Horst - Il libro dei circuiti Hi-Fi, L. 4.400
- ☐ 7 Bochum/Dögl - Guida illustrata al TVcolor service, L. 4.400
- ☐ 8 Schneider - Il circuito RC, L. 3600
- ☐ 9 Sehrig - Alimentatori con circuiti integrati, L. 3.600
- ☐ 10 Mende - Il libro delle antenne: la teoria, L. 3.600
- ☐ 11 Horst - Elettronica per film e foto, L. 4.400
- ☐ 12 Sutaner/Wissler - Il libro dell'oscilloscopio, L. 4.400
- ☐ 13 Wirsum - Il libro dei miscelatori, L. 4.800
- ☐ 14 Link - Metodi di misura per radioamatori, L. 4.000
- ☐ 15 Mende - Il libro delle antenne: la pratica, L. 3.600
- ☐ 16 B.U. Lewandowski - Progetto e analisi di sistemi, L. 3.600

biblioteca tascabile elettronica

- ☐ 1 Siebert - L'elettronica e la fotografia, L. 2.400
- ☐ 2 Zierl - Come si lavora con i transistori, parte prima, L. 2.400
- ☐ 3 Stöckle - Come si costruisce un circuito elettronico, L. 2.400
- ☐ 4 Richter - La luce in elettronica, L. 2.400
- ☐ 5 Zierl - Come si costruisce un ricevitore radio, L. 2.400
- ☐ 6 Zierl - Come si lavora con i transistori, parte seconda, L. 2.400
- ☐ 7 Tünker - Strumenti musicali elettronici, L. 2.400
- ☐ 8 Stöckle - Strumenti di misura e di verifica, L. 3.200
- ☐ 9 Stöckle - Sistemi d'allarme L. 2400
- ☐ 10 Siebert - Verifiche e misure elettroniche, L. 3.200
- ☐ 11 Zierl - Come si costruisce un amplificatore audio, L. 2.400
- ☐ 12 Baitinger - Come si costruisce un tester, L. 2.400
- ☐ 13 Gamlich - Come si lavora con i tiristori, L. 2.400
- ☐ 14 Zierl - Come si costruisce un telecomando elettronico, L. 2.400
- ☐ 15 Müller - Come si usa il calcolatore tascabile, L. 2.400
- ☐ 16 Biebersdorf - Circuiti dell'elettronica digitale, L. 2.400
- ☐ 17 Frahm/Kort - Come si costruisce un diffusore acustico, L. 2.400
- ☐ 18 Baitinger - Come si costruisce un alimentatore, L. 3.200
- ☐ 19 Stöckle - Come si lavora con i circuiti integrati, L. 2.400
- ☐ 20 Stöckle - Come si costruisce un termometro elettronico, L. 2.400
- ☐ 21 Zierl - Come si costruisce un mixer, L. 2.400
- ☐ 22 Zierl - Come si costruisce una radio FM, L. 2.400
- ☐ 23 Schiersching - Effetti sonori per il ferromodellismo, L. 2.400

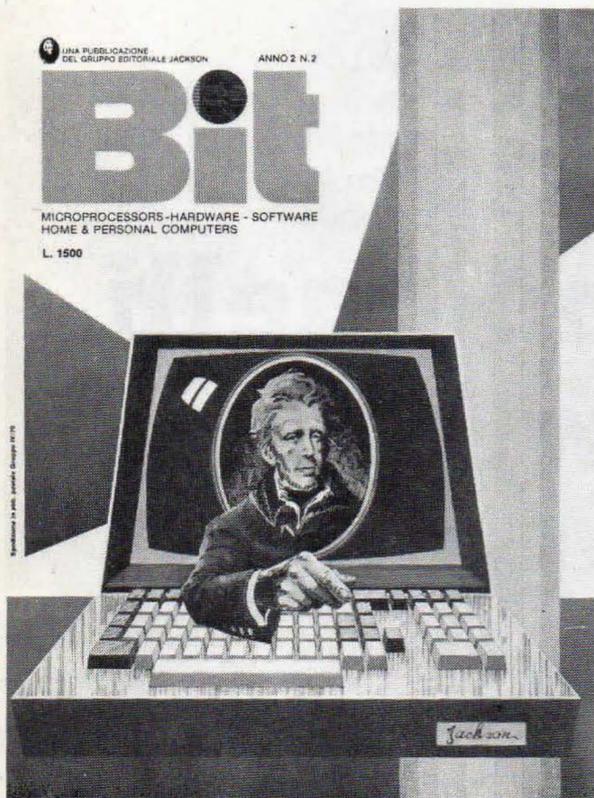
Tagliando da compilare, ritagliare e spedire in busta chiusa o incollato su cartolina postale a:

SELEZIONE
Via dei Lavoratori, 124
20092 CINISELLO B. (MILANO)

Prego inviarmi i seguenti volumi. Pagherò in contrassegno l'importo indicato più spese di spedizione.

________________ nome
________________ cognome
________________ indirizzo
________________ località
________________ c.a.p.
________________ codice fiscale

SOMMARIO

VOL. 2 N° 2 - Marzo 1979 - Maggio 1979

In copertina:
La Jackson Italiana e il Personal Computer.
Grafica di Renato NISI.

INDICE INSERZIONISTI

DIRETTORE RESPONSABILE:
Giampietro Zanga

GRAFICA E IMPAGINAZIONE:
Job Line srl

CONTABILITÀ:
Franco Mancini, Mariella Luciano

DIFFUSIONE E ABBONAMENTI:
Gabriella Napoli, Silvia Decari

DIREZIONE, REDAZIONE:
P.le Massari, 22 - 20125 Milano
Telefoni 68.03.68 - 68.00.54

AMMINISTRAZIONE:
Via Vincenzo Monti, 15 - 20123 Milano

PUBBLICITÀ: Concessionario per l'Italia e l'Estero Reina & C. S.n.c. - P.le Massari, 22/A - 20125 Milano
Telefoni 60.35.15 - 69.04.91

Autorizzazione del Tribunale di Milano n. 445 del 16/12/1978

FOTOCOMPOSIZIONE: Tip. Coop. Parole Nuove - Brugherio

STAMPA: Litografia del Sole srl - Buccinasco

Concessionario esclusivo per la diffusione in Italia e all'Estero:
SODIP - Via Zuretti, 25 - 20125 Milano
Via Serpieri, 11/5 - 00197 Roma

Spedizione in abbonamento Postale Gruppo III/70

Prezzo della rivista L. 1.500

Numero arretrato L. 2.500
Abbonamento annuo L. 6.000
per l'Estero L. 9.000

I versamenti vanno indirizzati a: Jackson Italiana Editrice S.r.l. Via Vincenzo Monti, 15 - 20123 Milano
mediante emissione di assegno bancario, o cartolina vaglia.
Per i cambi d'indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

Per evitarvi di andare a Boston, Dallas, New York, Tokio, San Francisco, Londra

abbiamo aperto a Cinisello Balsamo un computer shop come quelli di Boston, Dallas, New York, Tokio, San Francisco, Londra.

L'elaboratore per tutte le esigenze e alla portata di tutti. L'elaboratore per il dirigente, il professionista, il ricercatore, l'amatore.

Questo è quello che trovate alla Computeria - Il Centro del Personal Computer.

Alla Computeria sono rappresentati i piú importanti costruttori del settore. Potete cosí finalmente rendervi conto della differenza tra una casa e l'altra, tra uno strumento e l'altro. E scegliere quello che va bene per voi: con l'aiuto dei nostri tecnici.

Ma la Computeria non vi offre soltanto degli strumenti e la consulenza per aiutarvi nel loro acquisto. Vi offre un esperto servizio di assistenza tecnica sull'hardware e sul software applicativo, se e quando vi serve. E vi offre un programma completo di addestramento, utilissimo se è la prima volta che vi avvicinate all'elaboratore.

Uno dei prossimi giorni, fate un salto alla Computeria: cosí, per curiosare. Oltretutto, non avete neppure bisogno di prendere un jet.

La Computeria è nella nuova zona commerciale di Cinisello Balsamo, vicino allo svincolo di Viale Zara dell'autostrada A4, dietro allo Shopping Center e alla Rinascente.

Arrivederci presto.

20092 Cinisello Balsamo (Milano)
Palazzo Testi, Via Cantù 20
Tel. 02/6173032 - 6173150

EDITORIALE

Noi, voi e loro

È abbastanza evidente chi siamo noi: la Jackson Italiana, una piccola casa editrice che, sopravvissuta felicemente all'esplosione interna che ne ha fatto una grossa casa editrice, insiste ad usare ed a perseguire quei metodi che le hanno permesso di affermarsi, come la cura della completezza dell'informazione, la correttezza professionale, il dialogo con i lettori e qualche altra schiocchezza di questo tipo (perdonateci l'immodestia, lo sappiamo che è odiosa; in realtà si tratta di semplice orgoglio professionale).

È altrettanto chiaro chi siete voi, nel senso che voi che ci state leggendo in questo momento lo sapete benissimo; noi però non vi conosciamo ancora bene, ed a questo sarà meglio por rimedio al più presto; in realtà di voi non conosciamo nulla, salvo il profumo del vostro denaro (o nulla del tutto, se la rivista l'avete presa dalla scrivania del collega invece di comperarla). A qualcuno questo può bastare, a noi no!

Ci limitiamo ad immaginare chi potete essere: persone concrete, cui i discorsi elevati di marketing provocano incontenibili accessi di sonno; tecnici e ricercatori, più avvezzi al come, quando, dove e perchè che non al non a caso o al nella misura in cuit pilli che di Elettronica Oggi leggono quasi soltanto la pubblicità, perchè a loro dire è l'informazione più concisa ed attendibile dell'intera rivista (non è vero, lo sapete benissimo anche se vi divertite a punzecchiarci ed a stuzzicare il nostro self-control); oppure appassionati di elettronica logica, che in questa fuggono per dimenticare un lavoro noioso che quotidianamente fa del suo meglio per affogarli nel nulla. In una parola, drogati dell'elettronica, uomini che talvolta, sopra pensiero, in una limpida notte di luna, sussurrano teneramente alla ragazza qualcosa che sa di fan-out, di bus e di left shift. Se è così, siete dei nostri.

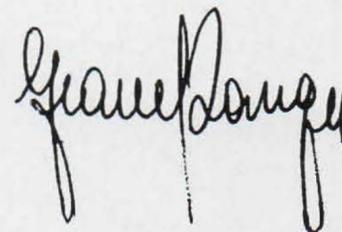
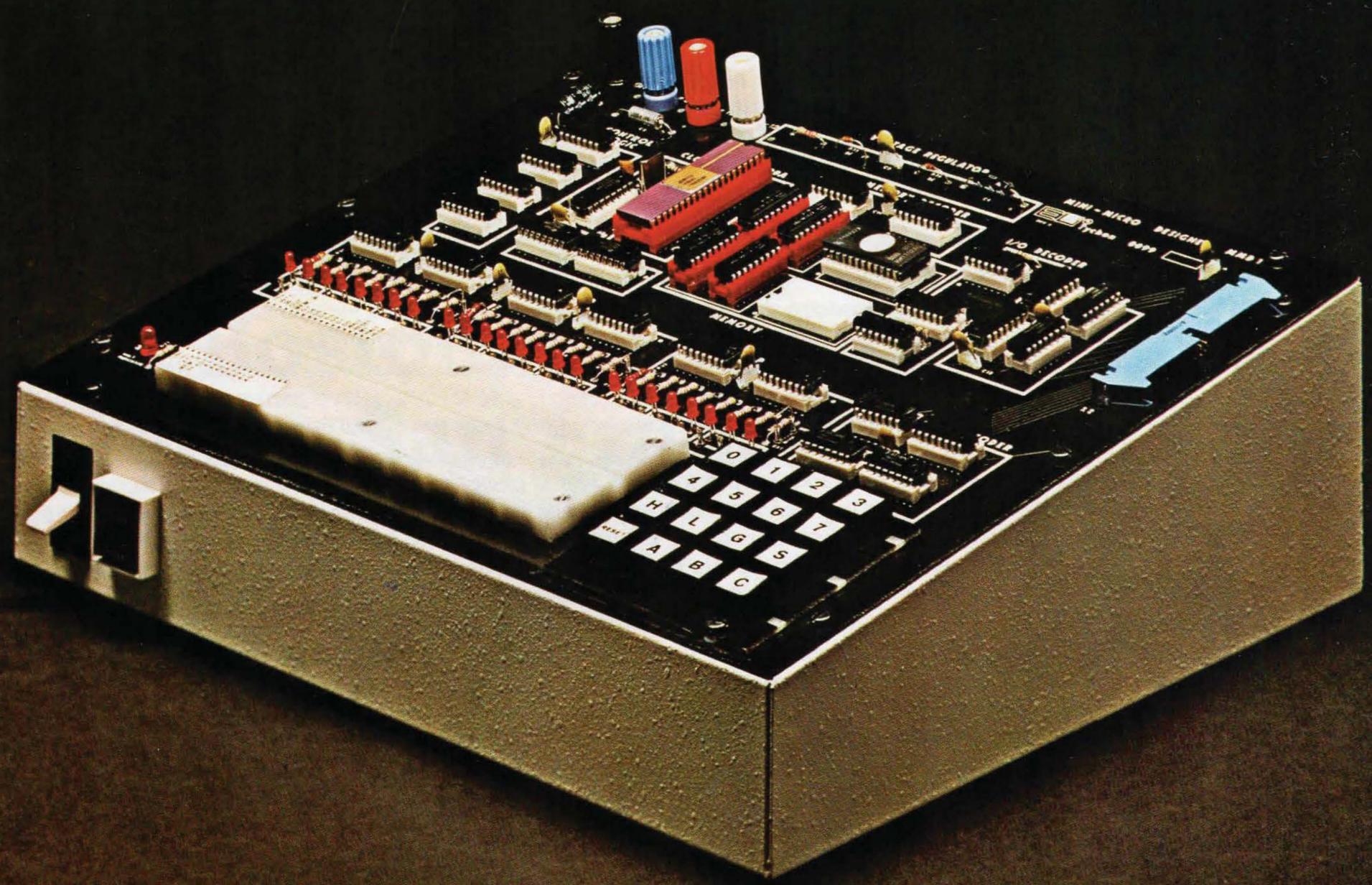
Siete dei nostri se vi mette a disagio un interlocutore che si balocca in chiacchiere senza concludere alcunchè di comprensibile, o se vi mandano in bestia tutte quelle difficoltà dovute esclusivamente alla stupidità umana («... sì, il pezzo lo abbiamo, ma per ragioni di amministrazione non lo possiamo vendere che al martedì...»), o se dovete tenere a bada l'incombente follia omicida che si sviluppa in voi davanti alle difficoltà che loro mettono sul vostro cammino senza badare particolarmente al fatto che la cosa potrebbe non essere di vostro gradimento, tanto non gli interessa («...sì, abbiamo preso in considerazione la sua richiesta di impiego; ora riempia da bravo questo formulario e entro sei mesi le faremo sapere qualcosa...»).

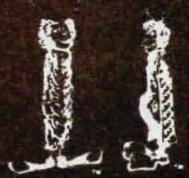
Già, loro. Anzi, Loro, perchè spesso sono potenti. Non è chiaro il perchè, forse è solo un'impressione, ma spesso pare che il potere renda mediocri, o che la mediocrità renda potenti; fatto si è che chi si occupa di chiacchiere tende ad avere la via molto più sgombra di chi si occupa di fatti, e tante meno idee contengono le chiacchiere, tanto più il loro titolare arriva in alto. Almeno qui ed ora; speriamo che si tratti di un fatto transitorio. Ma chi sono poi loro? Bah, non è facile dirlo; in generale vi accorgete che una persona è uno di loro quando, valendo meno di un decimo di quanto valete voi, ha su di voi un grosso potere, lo esercita senza risparmio e ne gode. Chiunque può essere dei loro, un genitore, un insegnante, un superiore, un collega, molti burocrati e politici: persone mediocri che si avvalgono appunto della mediocrazia circostante per cavarne un rispetto ed un potere che nessuno si sognerebbe di attribuirgli.

Ma, in omaggio al vecchio proverbio cinese (se hai un nemico, siediti in riva al fiume e aspetta; prima o poi vedrai passare il suo cadavere), conviene avere pazienza: loro sono molto sensibili ai cambiamenti, perchè raramente sono abbastanza elastici da superarli (infatti li temono più della peste e, quando possono, li ostacolano con tutti i mezzi); l'elettronica è per sua natura una scienza in rapida espansione, che provoca e provocherà moltissimi cambiamenti sia al suo interno, che al suo esterno. Quindi, teniamo duro e non lasciamoci scoraggiare; parliamone invece fra noi, molte volte consola e spesso è utile.

A questo proposito servitevi pure di BIT: se qualcuno di loro vi ha fatto uscire dai gangheri, se vi ha truffato tanto bene da non poter far valere in alcun modo le vostre ragioni, se vi mette i bastoni fra le ruote senza alcun motivo, se non ne potete più e se il fatto ha anche una vaga attinenza col mondo dell'elettronica digitale (succede molto più spesso di quanto non si possa sospettare), scriveteci ed avrete tutta la soddisfazione che possiamo darvi. Scriveteci anche se avete qualche buona idea, sia su argomenti elettronici, sia sul come dovremmo organizzare la rivista perchè vi piaccia di più: noi dipendiamo da voi molto più di quanto voi dipendiate da noi.

È tutto, per ora, di chiacchiere qui ne abbiamo fatte fin troppe.

Giampietro Zanga

IL MICROGIGANTE DELL'ISTRUZIONE

MMD1

IL MICROCOMPUTER DEI BUGBOOKS

CPM Studio

Nuova era: primo passo

Il computer sta avviando una nuova rivoluzione industriale: la distanza tra sistemi computerizzati e utenti si sta assottigliando con una rapidità che neppure il più ottimista degli anni '60 avrebbe potuto prevedere!

Il punto di decollo sono stati i circuiti L.S.I. (Large Scale Integration), che hanno reso possibile il trattamento economico di una entità che ha sempre accompagnato lo sviluppo umano: l'informazione.

Dal Computer si è giunti al Personal Computer.

Non è ancora l'ideale, ad esempio, per quanto concerne la memorizzazione economica di grandi quantità di dati, ma già su questo punto si è sulla buona strada: dispositivi completamente elettronici come le memorie CCD (Charge-Coupled Devices) e le bolle magnetiche per un verso insidiano le posizioni occupate dalle memorie di massa attuali, per l'altro indicano nuove fasce di applicazioni.

Non è ancora l'ideale per quanto riguarda l'interazione utente-calcolatore (nel senso del non specialista che fa da sé i propri programmi), mediata attualmente da strumenti di programmazione che, sviluppati per far fronte all'hardware degli anni '50, risultano già inadeguati per le esigenze attuali: ma anche su questo punto c'è del nuovo. Il PASCAL sembra destinato a diventare il linguaggio di programmazione capace di realizzare nuovi sviluppi in ambito software.

Non è ancora l'ideale perchè un professionista, ad esempio un medico, lo utilizzi come schedario dei propri pazienti, o perchè venga utilizzato nell'ambito delle strutture scolastiche come strumento didattico, o perchè venga adoperato come un nuovo aggeggio, accanto al televisore o all'impianto HI-FI, per fornire un insieme di servizi domestici facilmente individuabili (bilancio mensile, rubrica telefonica, dieta, sistema didattico, agenda appuntamenti, calcolatrice, etc.): ma questo accade per varie ragioni, la prima delle quali è senz'altro l'impreparazione ed il non supporto di un'utenza completamente diversa da quella usuale.

Tutti questi elementi giustificano l'affermazione di nuova era dei sistemi computerizzati.

Comunque già più sensibile è l'interesse dell'utenza classica, quella della elaborazione dati, verso questi sistemi di costo decisamente inferiore rispetto ai mini: è auspicabile solo che la domanda, proveniente soprattutto dalla piccola azienda, non venga ricacciata indietro da una situazione non ancora preparata al supporto di tale problematica.

LA **HOMIC** PRESENTA

Home microcomputer s.r.l.
v. Dante 9 - Milano - tel. (02) 809456

il TRS 80

omologato per la rete europea
220 Volt - 50 Hz

Il TRS 80 è l'ideale per contabilità di piccole e medie aziende, per la didattica, per applicazioni scientifiche, per ogni uso personale.

Del TRS 80 sono già disponibili le seguenti espansioni:

★ Minifloppy da 89.600 bytes, velocità di trasferimento 125.000 bits/sec

★ Stampante ad aghi di tipo professionale da 80 o 132 colonne, velocità da 60 a 110 caratteri/sec

★ Espansioni di memoria per giungere sino a 32 K di RAM

IL TRS 80 VIENE VENDUTO NELLE SEGUENTI CONFIGURAZIONI

Memoria RAM	Sistema operativo	Prezzo
4 K bytes	Basic Level 1° 4 K (ROM)	**L. 1.088.000 + IVA**
16 K bytes	Basic Level 1° 4 K (ROM)	**L. 1.561.000 + IVA**
4 K bytes	Basic Level 2° 12 K (ROM)	**L. 1.267.000 + IVA**
16 K bytes	Basic Level 2° 12 K (ROM)	**L. 1.776.000 + IVA**

NEWSLETTER

GIM: un sistema a due μP

Mentre personal computers e videogames programmabili si fondono in un prodotto chiamato home computer, i costruttori di hardware si pongono un problema: come fare per rendere queste macchine facili da usare ma nello stesso tempo capaci di svolgere qualcosa di più di un gioco per quanto «pensato» e complesso possa essere. Una risposta viene dalla General Instrument (Microelectronics division) che ha progettato un sistema basato non su una, bensì su due microprocessori: un CP 1610 a 16 bit, per il controllo di tutti i blocchi funzionali (RAM, ROM, generatori di suoni) e un AY-3-8900 a 14 bit per il controllo della grafica a colori. Il sistema è in due versioni: la prima, quella di base, è programmata con cassette ROM simili a quelle usate per giochi programmabili; l'altra presenta più possibilità. In entrambi i casi per il display bisogna rifarsi ad un televisore.
La GIM non venderà direttamente questi sistemi, i cui prezzi molto prudenzialmente possono stimarsi in un range compreso fra 150-400 dollari, ma demanderà ad altri questo compito, assumendosi invece l'assistenza. Le cassette programmate, disponibili a 5 $, saranno garantite contro imitazioni. A tal fine è stato raggiunto un accordo con la inglese EMI Ltd, in possesso di particolari processi per la «filigranatura» delle cassette.

Alleanze in vista

La prima a lanciarsi è stata la ITT (duraturo accordo commerciale con la Apple Computer). Ora altre multinazionali dell'elettronica programmano di fare lo stesso per non trovarsi un domani disarcionate. Come investimento di medio-lungo periodo il personal computer sembra interessare i magnati della grossa industria. L'entità dell'investimeno è il parametro della scelta decisionale. Due esempi: in buona liquidità la Siemens è per l'acquisizione di una azienda americana del settore, iniziativa peraltro congeniale alla sua politica di investimenti negli USA. La francese Thomson-CSF è per un accordo meno impegnativo: produzione su licenza, produzione e distribuzioni esclusive in Europa, accordo tecnico/commerciale.

Incertezze di nome Texas Instruments

Nella sua annuale disamina sul mercato dell'elettronica professionale e consumistica Electronics esprime la seguente considerazione sugli home computers: pochi venditori sono disposti ad avanzare previsioni sull'andamento commerciale di questi sistemi. Sul loro mercato pesa l'attesa delle novità preannunciate dalla Texas Instruments che quando si muove, lo si sa, provoca impatti notevoli e ripercussioni a catena. Un anno critico quindi per gli home computers? Il problema è di arrivare al consumatore medio e all'imprenditore medio, raggiungibili concentrandosi sul software e sulla standardizzazione delle interfacce con dispositivi I/O.
Statisticamente il consumo americano di home computers, secondo Electronics, da 10 milioni di dollari del 77 e dai 21 milioni di dollari del 78 dovrebbe salire quest'anno a 34 milioni e a 65 milioni di dollari nel 1982.

Fra casa ed azienda chi vincerà?

È ancora prematuro per tentare di fare il punto sul mercato dei personal computer e delineare con sufficiente convinzione e serietà i suoi orientamenti. Ciò vale particolarmente per l'Europa ma nella sostanza anche per gli USA. Questa la premessa fatta dalla Creative Strategies ad una indagine compiuta su tale mercato, indagine dalla quale sono stralciati i due prospetti riportati. L'inchiesta ha permesso di quantificare in 43.000 i sistemi venduti nel 77 negli USA e di stimare in 1,2 milioni di unità il prevedibile consumo nel 1987 (da 172 a 3.500 milioni di $ la variazione in valore). La Creative Strategies ha considerato come personal computer quei sistemi di un costo inferiore a 10.000 $, comprensivo di periferici opzionabili e di un minimo di software applicativo. Due le categorie o correnti di prodotti che formano il mercato: gli home computer e i computer per usi differenti da quelli casalinghi. Nell'ambito del primo segmento il convincimento degli esperti della CS è che le prospettive migliori nel medio periodo vadano ai sistemi acquistati per impieghi di natura consumistica, tipo videogames sofisticati. Nell'ambito del secondo e più importante segmento almeno tre tipi di applicazioni gareggiano per la prima posizione.

A) **Il mercato americano degli home computers (migliaia di unità)**

	78	**79**	**80**	**81**	**82**
Hobbystica	35,1	45	50	45	30
Home programmers	80,2	145	210	260	290
Consumer	24,0	75	140	220	310

B) **Il mercato americano dei calcolatori personali diversi da quelli per la casa (migliaia di unità)**

	78	**79**	**80**	**81**	**82**
Industria	5,2	10,0	16	24	35
Scuola	23,4	35,0	46	63	105
Appl.ni scientifiche	17,8	37,0	60	100	140
Appl.ni professionali	5,0	9,5	15	30	50
Low-end business	35,2	142,0	190	190	140
Very small business	17,0	26,5	39	85	143

Il miglior canale...

Stando ai risultati di una breve inchiesta pubblicata da Electronics News il miglior sistema per raggiungere il consumatore finale di personal ed home computer è ancora da trovare e chiarire. Nel 1977 quasi la metà dei sistemi è affluita a destinazione attraverso i cosidetti «retail stores»; nello scorso anno il mercato ha già mostrato segni di specializzazione: all'incirca la metà ha trovato alimento dai «specialty retail outlets», negozi specializzati.
E per il resto? Tutte le maniere son buone per vendere, almeno fino a quando industria e commercio non avranno trovato o concordato un assetto definitivo.

Thank you government

Nei prossimi due-tre anni il governo inglese programma di spendere per la fertilizzazione delle attività applicative inerenti i microprocessori diverse decine di miliardi di lire. L'azione divulgativa ed informativa si rivolge soprattutto alle piccole-medie aziende. Tra gli altri effetti tale politica dovrebbe anche creare le premesse per una esplosione del personal-computing. Pienamente convinti di questo fenomeno sono i responsabili della Nascom Microcomputer Ltd. Da diversi anni questa azienda tratta piastre e prodotti affini e complementari. Ora ritiene che con l'arrivo di piastre a basso costo - come la VIM della Synertek, la Superboard 2 della Abacus Computers Ltd ed altri kit della Mostek e della Zilog, unitamente agli sforzi divulgativi del governo, il mercato possa prendere uno slancio considerevole. Per dare maggiore consistenza allo sbocco essa progetta varie innovazioni a livello di software e di unità opzionabili.
Sicura della svolta che assumera il mercato, la Nascom ha passato alla Mostek un ulteriore ordine di piastre ed altri kit per 1,5 milioni di dollari.

I giapponesi non spaventano

Sharp e Casio, sono al momento, i soli gruppi giapponesi di elettronica che producono e vendono personal computers. Ma ancora per poco. Sanyo, Hitachi, Toshiba e Matsushita, per restare alle più importanti case, hanno tutte pronti dei prototipi e sono quindi in grado di entrare nel vivo del mercato appena vedranno una schiarita che prometta di durare. Invero pare che solo la Toshiba abbia un sistema capace di competere con i prodotti americani.
Ciò spiega le defaillances e le perplessità manifestate dai nipponici in occasione dell'americana Winter Consumer Electronic Show. Pare proprio che per ora e in questo settore i giapponesi non costituiscono una preoccupazione!

A scacchi con il calcolatore

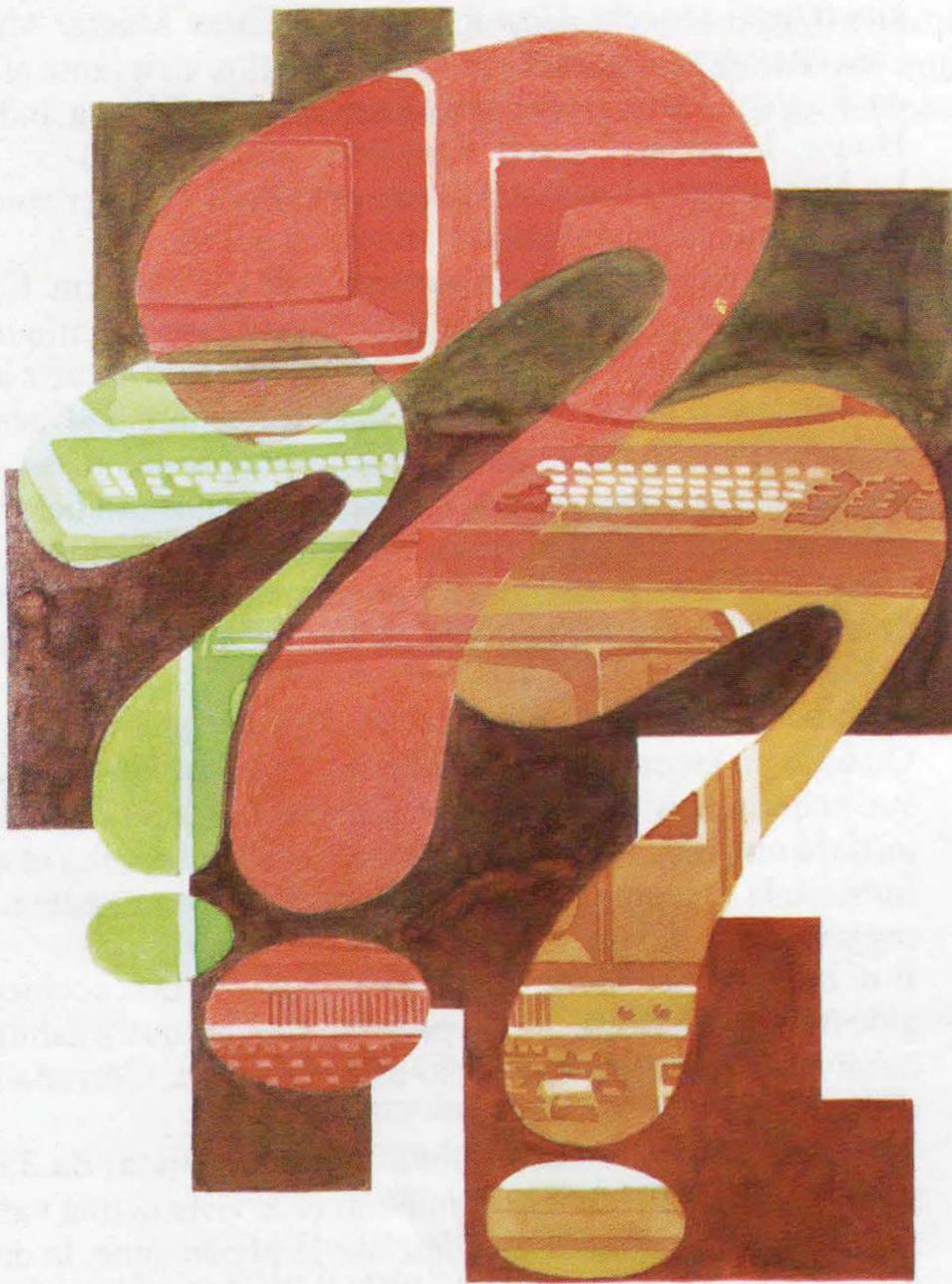
L'entrata di Hong Kong nel campo dei calcolatori adibiti ai giochi di società, acquista nuovi punti con l'introduzione di un nuovo piccolo calcolatore per il gioco degli scacchi. Con il nome di «Chess Master Mark 1», questo apparecchio che copre sei gradi di difficoltà (dal principiante al professionista più freddo e calcolatore), è una realizzazione della Novag Industries Ltd. (1417 Star House, Harbour Centre, Kowloon, Hong Kong).
La Novag è stata avviata nel maggio del 1978, per studiare, fabbricare ed immettere sul mercato questo nuovo prodotto.
Il «Mark 1» è un piccolo apparecchio di 12x22x5 cm. È semplice da usare. Non è invece altrettanto semplice tutto il processo del minuscolo cervello elettronico, programmato in modo di essere al corrente di tutte le regole internazionali.
Per iniziare il gioco lo sfidante deve informare il «Chess Master» del livello di diffiçcoltà scelto per la partita. Decisa la prima mossa, il giocatore deve impostare l'informazione corretta, premendo i pulsanti del pannello frontale del calcolatore. Con questa operazione la mossa viene riportata sul display. La macchina dà la contromossa entro pochi secondi o anche dopo alcune ore, a seconda della difficoltà scelta per la partita (la macchina programma anche le mosse successive).
Il gioco procede fino allo scacco matto o alla fine per parità.
Quando un giocatore vince, la macchina illumina la parola «lose» (perdente) sul suo schermo. A questo punto si può iniziare una nuova partita. Si può anche iniziare una partita da metà gioco o verso la fine, ma in questo caso bisogna informare la macchina delle posizioni di ciascuna pedina. È anche possibile correggere delle mosse sbagliate.
Il «Chess Master Mark 1» non è venduto con la scacchiera, la maggior parte dei giocatori preferisce usare la propria. Comunque è fornito di un riduttore AC o di un altro per l'eventuale uso in automobile. Correda il sistemino un libretto d'istruzioni.
La produzione di questo nuovo gioco è iniziata: da 350 a 500 apparecchi al giorno. Tale cifra dovrebbe aumentare in vista di una vasta campagna pubblicitaria. Ancora prima di aver iniziato la produzione, la ditta aveva già prenotati 65.000 apparecchi destinati all'Europa e al Medio Oriente.
La ditta sta cercando degli agenti nei vari potenziali mercati verso i quali pensa di esportare l'apparecchio a circa 95.000 lire FOB, un prezzo particolarmente interessante rispetto ad altri sistemi simili provenienti dagli USA.

Due calcolatori dalla Atari

Era da un anno che la Atari, una delle prime aziende a buttarsi sul mercato dei videogames, andava dicendo di voler intraprendere una attività di personal computers. Finalmente nello scorso dicembre la società è passata alle vie di fatto con l'annuncio dell'Atari 400 e dell'Atari 800, due sistemi programmabili dall'utente con linguaggio Basic (altri linguaggi di programmazione saranno disponibili su cassette pre-registrate). Qualche dato. L'Atari 400 è un sistema general purpose progettato per aiutare i consumatori a convertirsi dai videogames ai calcolatori personali. La sua tastiera consta di 57 tasti e la memoria massima arriva a 8000 bytes. Orientativamente il suo costo può stimarsi in 400 dollari. L'Atari 800 è molto più potente (48.000 bytes di memoria su RAM) e risulta particolarmente predisposto per essere collegato a periferiche (stampanti, floppy disk, etc.) Costa il doppio del precedente modello.

NUCLEO

Introduzione al computer

a cura della Redazione

Introduzione

È ormai acquisito che in questi ultimi anni l'industria elettronica ha subito una profonda innovazione a causa dell'avvento dei circuiti realizzati con tecnologia L.S.I. (*Large Scale Integration*).

Questi sono circuiti contenenti su un piccolo rettangolo di silicio (cioè su di un *chip*) migliaia di elementi attivi (transistors) e passivi (resistori e condensatori) (Fig. 1).

Tra i componenti L.S.I. quello che è risultato determinante è stato il microprocessore, che è per l'appunto un circuito L.S.I. che implementa le principali funzioni di un computer tradizionale. Non è quindi un caso che, insieme alle numerosissime applicazioni industriali, anche l'industria del computer abbia tratto beneficio dall'avvento di questa tecnologia, quanto meno per la drastica riduzione di prezzo e di dimensioni delle parti tradizionalmente costose di questi sistemi.

La situazione attuale è pertanto sorprendente: costa meno il computer vero e proprio dei dispositivi mediante i quali si comunica con esso, ossia le *periferiche*, quali stampanti, terminali, video, etc.

I sistemi definiti hobby o personal o business computers sono per l'appunto il risultato ultimo a cui è arrivato il «computer».

Anche la filosofia sul computer è significativamente cambiata. Fino agli anni 60, in conseguenza dei notevoli costi, della complessità dei circuiti, della dimensione notevole dei computers, erano individuate come *uniche* utenze possibili quelle rappresentate dai grossi enti pubblici o privati, in quanto *le sole* che potevano disporre dei mezzi finanziari necessari all'acquisizione di tali sistemi. Inoltre erano richiesti operatori estremamente specializzati, e infine il luogo dove le informazioni erano elaborate era distinto dal luogo in cui era l'utente.

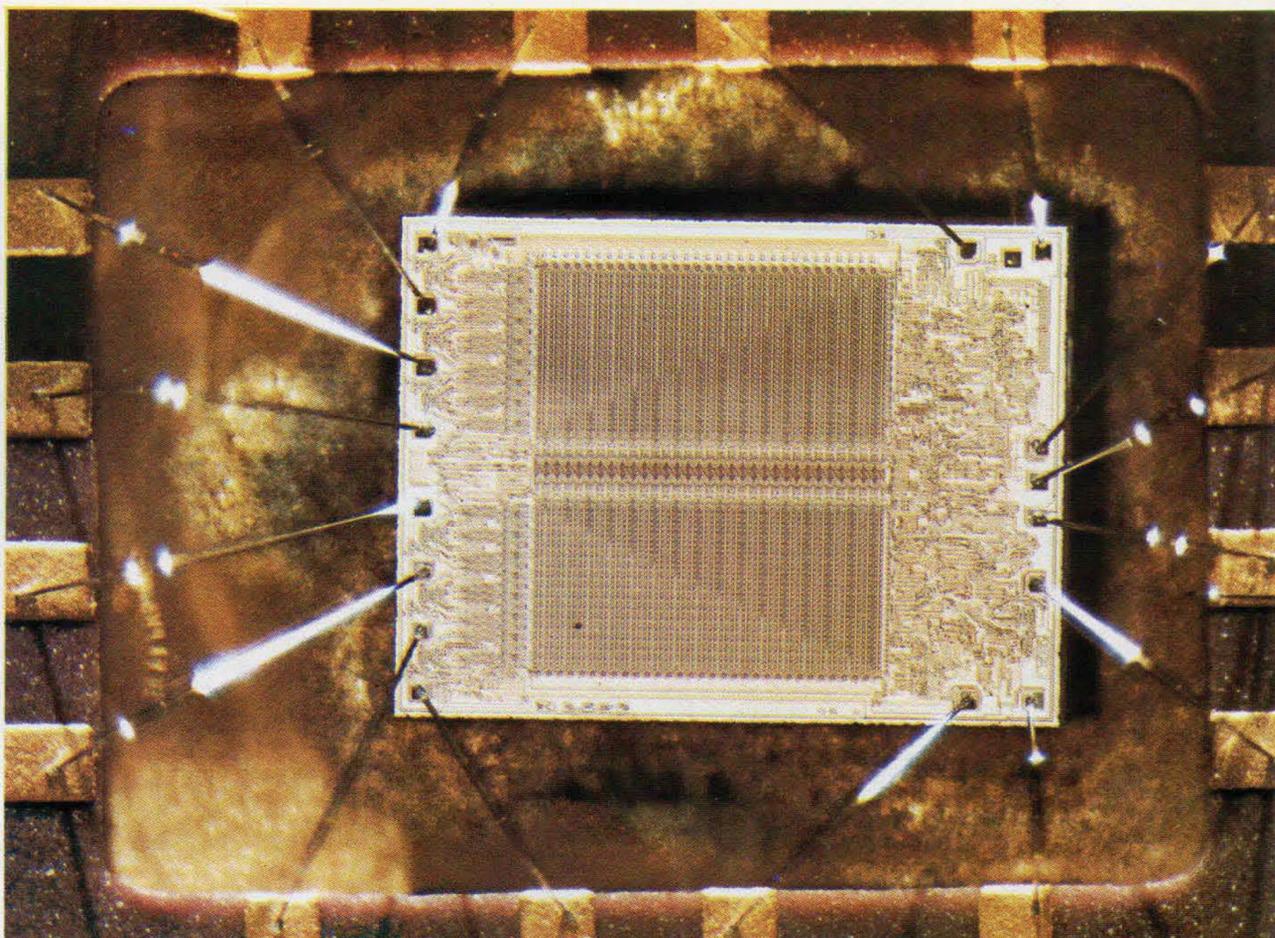



Figura 1 —Chip. Le misure sono 2,54x3,30 mm.

Oggi è possibile affermare il contrario: *un personal computer non è altro che un computer che costa poco, di dimensioni limitate, utilizzabile da chiunque, utilizzabile per qualunque scopo, potente.*

Che cosa è un personal computer?

Un personal computer è, dunque, un computer ovvero un dispositivo capace di eseguire una sequenza memorizzata di istruzioni, cioè di ordini. In questo modo il computer è in grado di svolgere le funzioni per le quali è stato programmato, per quanto complesse siano. Le istruzioni che un computer esegue *realmente* sono, comunque, comandi molto elementari che hanno scarsa attinenza con una formulazione per quanto semplificata di un generico problema. L'insieme di questo tipo di istruzioni o comandi è chiamato il *linguaggio macchina* del computer.

Per fortuna è anche possibile fornire al computer istruzioni più sofisticate, ossia più aderenti ad una formulazione passo passo di un problema: in questo caso il linguaggio utilizzato per programmare il computer è chiamato *linguaggio ad alto livello* (il BASIC è uno di questi).

Per comprendere meglio che cosa è un computer, non rimane che descriverne la struttura.

Fondamentalmente si distinguono le seguenti unità funzionali (fig. 2):

- Unità di calcolo/controllo ovvero C.P.U. *(Central Processing Unit)*
- Memoria principale
- Logica di governo delle periferiche di ingresso
- Logica di governo delle periferiche di uscita.

C.P.U.

La C.P.U. esegue le istruzioni macchina prelevate una alla volta dalla memoria principale. Ha pertanto due funzioni principali: 1) prelevare nella sequenza giusta le singole istruzioni dalla memoria; 2) eseguirle sfruttando le proprie capacità aritmetico-logiche.

Queste funzioni sono tipicamente implementate da un singolo circuito L.S.I.: il microprocessore (fig. 3).

Memoria

Contiene la sequenza di istruzioni da eseguire, ossia il *programma in esecuzione,* e i *dati* su cui operare.

Funzionalmente è organizzata in *parole,* che rappresentano l'unità logica di informazioni che la C.P.U. è in grado di acquisire in modo diretto.

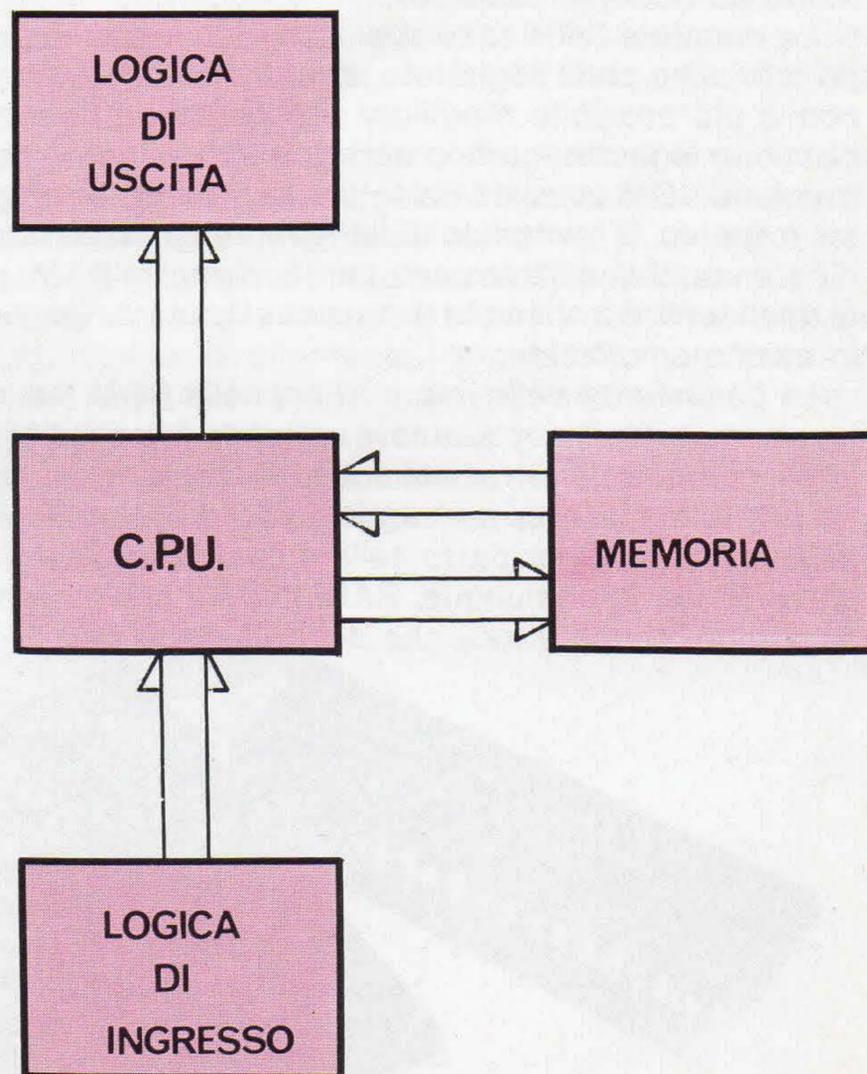



Figura 2 — Schema funzionale di un microcomputer.

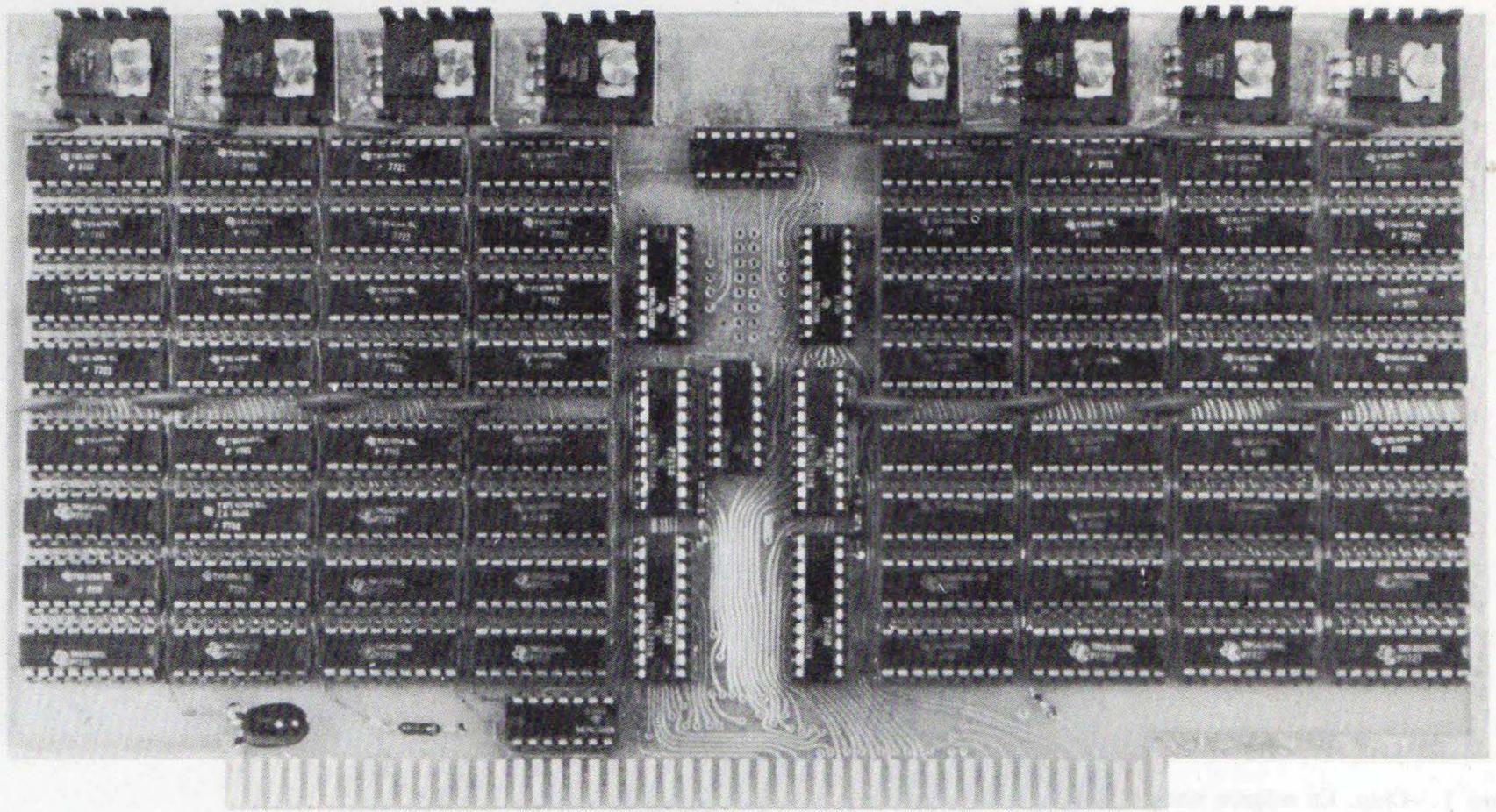



Figura 4 — Scheda di memoria RAM.

La memoria principale, nei sistemi a microprocessore, è implementata con componenti L.S.I. I tipi fondamentali sono:

memoria ROM (*Read Only Memory* = memoria a sola lettura)

memoria RAM (*Random Access memory* = memoria ad accesso casuale).

Le memorie ROM sono così chiamate in quanto, dopo che sono state registrate, scritte (dal costruttore), non è più possibile modificar e il contenuto, ossia è possibile leggerle ma non scriverle. Il vantaggio della memoria ROM consiste nel fatto che, quando il computer è spento, il contenuto della ROM resta inalterato, a differenza di quanto accade per le memorie RAM, per le quali invece avviene la distruzione delle informazioni in essa memorizzate.

La persistenza delle informazioni nella ROM è di notevole importanza, in quanto in tal modo è possibile fornire il computer di istruzioni che si desidera non vengano *mai* «dimenticate» per ragioni che saranno chiare in seguito. La maggior parte della memoria principale di un computer è, comunque, RAM (fig. 4), che permette operazioni sia di lettura che di scrittura.

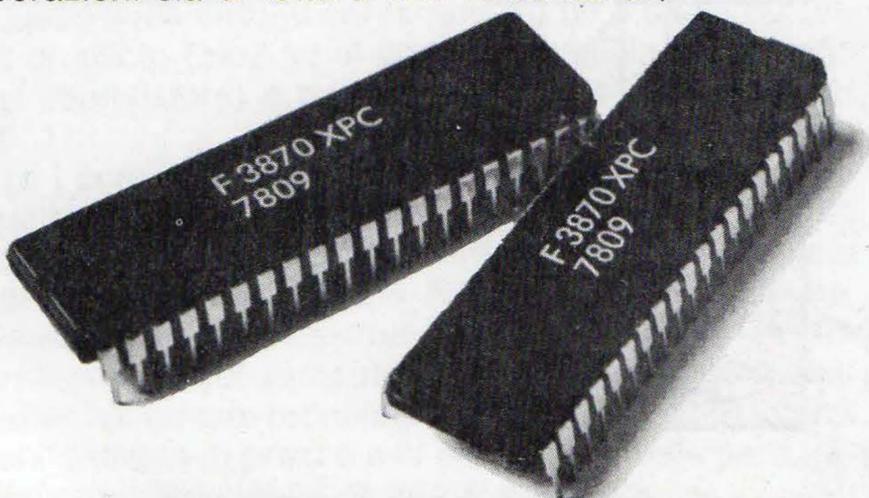



Figura 3 — Microprocessore.

Logica di governo di Ingresso/Uscita

Mediante queste unità è possibile comandare, controllare tramite la C.P.U. le periferiche connesse al computer, e realizzare lo scambio dati tra computer e mondo esterno.

In aggiunta alle unità fin qui sommariamente descritte, un microcomputer include l'alimentatore necessario a fornire al sistema le tensioni richieste.

Se la memoria richiesta è piccola, un microcomputer può essere implementato su una singola scheda. In questo caso si parla di computer su singola scheda, ovvero di SBC (*Single Board Computer*). In generale però si ha una scheda di C.P.U., una o più schede di memoria, una o più schede di I/O necessarie all'interfacciamento del computer con le opportune periferiche.

Hardware e Software

La situazione fin qui descritta costituisce l'*hardware* del computer. Perchè un computer compia elaborazioni, è necessario che esegua istruzioni, ossia che sia fornito di software. *Software* è, pertanto, l'insieme di programmi associato al computer.

Il programma che il computer esegue è presente nella memoria del sistema. L'esecuzione di un programma consiste nell'esecuzione delle singole istruzioni che lo costituiscono; l'esecuzione delle singole istruzioni avviene, in maniera semplificata, nel seguente modo:

a) la C.P.U. legge dalla memoria l'istruzione da eseguire;
b) la C.P.U. si predispone alla lettura, a fine ciclo, della istruzione successiva;
c) la C.P.U. decodifica l'istruzione letta nella fase a);
d) la C.P.U. esegue l'istruzione decodificata nella fase c);

e) la C.P.U. ritorna alla fase a), ricordando la fase b).

Quanto sopra evidenzia, tra l'altro, la ragione per cui il programma in esecuzione deve essere in memoria. In altre parole occorre tener presente che *la lunghezza del programma che si desidera eseguire è limitata dalla quantità di memoria RAM a disposizione del sistema.*

Software e Firmware

Il software che viene «congelato» in ROM viene chiamato firmware. Quindi il *firmware* non è altro che l'insieme dei programmi disponibili su memoria ROM invece che RAM.

La distribuzione dei programmi tra ROM e RAM non è casuale, ma dipende da una precisa ragione. Infatti un qualunque computer, se all'accensione non trova istruzioni da eseguire, non solo non fa nulla, ma neppure permette che possa essere istruito per far qualcosa. Pertanto un microcomputer *deve* avere residente in ROM un programma, eseguibile all'accensione, che permetta l'avvio del computer. Il tipo di avvio dipende dal tipo di programma eseguito all'accensione.

Un tipico programma che corrisponde a questi requisiti è il programma *Monitor* che, tramite comandi forniti da tastiera, permette di leggere e scrivere in memoria o nei registri della C.P.U., di avviare l'esecuzione di un piccolo programma caricato in memoria mediante i comandi di scrittura, etc. In firmware si trovano conseguentemente i programmi di gestione delle periferiche principali (*peripheral handler*).

Nei sistemi più completi, il programma eseguito all'accensione è il *programma di autocaricamento* (*autoload o bootstrap* in inglese) la cui funzione è di caricare nel computer il *software di sistema*, ossia quel tipo di software fornito dal costruttore per rendere estremamente agevole l'utilizzo del computer.

Infine nei computers che lavorano *solamente con un* linguaggio ad alto livello (ad esempio il BASIC), in firmware si troverà l'interprete.

Periferiche

Un computer deve essere fornito di dispositivi di ingresso/uscita perchè sia operabile. È possibile distinguere tre tipi di periferiche:

a) periferiche di ingresso
b) periferiche di uscita
c) memoria di massa.

La periferica di ingresso per eccellenza nei sistemi a microcomputer è la **tastiera**. Poichè i dati sono memorizzati nel computer come sequenza di Ø ed 1, ad ogni tasto corrisponde una particolare combinazione di Ø, 1, ossia ad ogni tasto corrisponde un codice particolare.

I caratteri alfanumerici sono codificati universalmente con una sequenza di 8 bit, cioè in byte (1 byte è formato da 8 bit ossia da 8 cifre binarie), per cui essendo $2^8 = 256$ le possibili combinazioni che si possono ottenere con 8 bit, è possibile codificare fino ad un massimo di 256 caratteri. Ora, il numero dei caratteri alfanumerici è notevolmente inferiore a 256, anche se è presa in considerazione la serie di caratteri speciali che compaiono nelle tastiere professionali. Pertanto per codificare i caratteri è utilizzato un sottoinsieme di tali combinazioni. Nel caso dei personal computers il codice adottato per la codifica dei caratteri alfanumerici è il codice ASCII (*American Standard Code for Information Interchange*).

Premendo allora un tasto, la tastiera fornisce la codifica in ASCII del carattere stesso, ed in più i relativi segnali di controllo. L'acquisizione del codice è quindi fatta dalla logica di governo di ingresso, che pertanto presenta alla C.P.U. il codice per l'elaborazione successiva (fig. 5).

I più usuali dispositivi di uscita nei sistemi microcomputer sono:

LED (*Light Emitting Diode*). Utilizzati soprattutto nei sistemi didattici permettono la visualizzazione di una singola cifra (Ø÷9) o di una lettera compresa tra A ed F.

CRT (*Cathode Ray Tube*). È lo schermo TV. Questo dispositivo è quello più diffuso nei sistemi personal-business computer, dato che soddisfa le esigenze di visualizzazione di pagine di dati in tempi veloci. Di solito la pagina visualizzata sullo schermo TV ha la propria immagine in RAM, per cui ad ogni cambiamento in RAM corrisponde l'immediato aggiornamento dello schermo. Le caratteristiche di linee/colonne visualizzate sono tipicamente da 16 a 32 linee e da 32 a 64 caratteri per linea.

Questi valori dipendono da come si è realizzata la connessione tra schermo TV e computer. Qualora ci si connetta direttamente all'antenna, si va incontro ad una riduzione della larghezza di banda, per cui 32/40 caratteri sono la massima capacità per linea. Utilizzando, invece, direttamente i segnali di sincronismo (saltando la sezione RF), è possibile disporre di una larghezza di banda maggiore e quindi di un maggior numero di caratteri per linea.

Stampante: è indipensabile quando, per il tipo di ap-

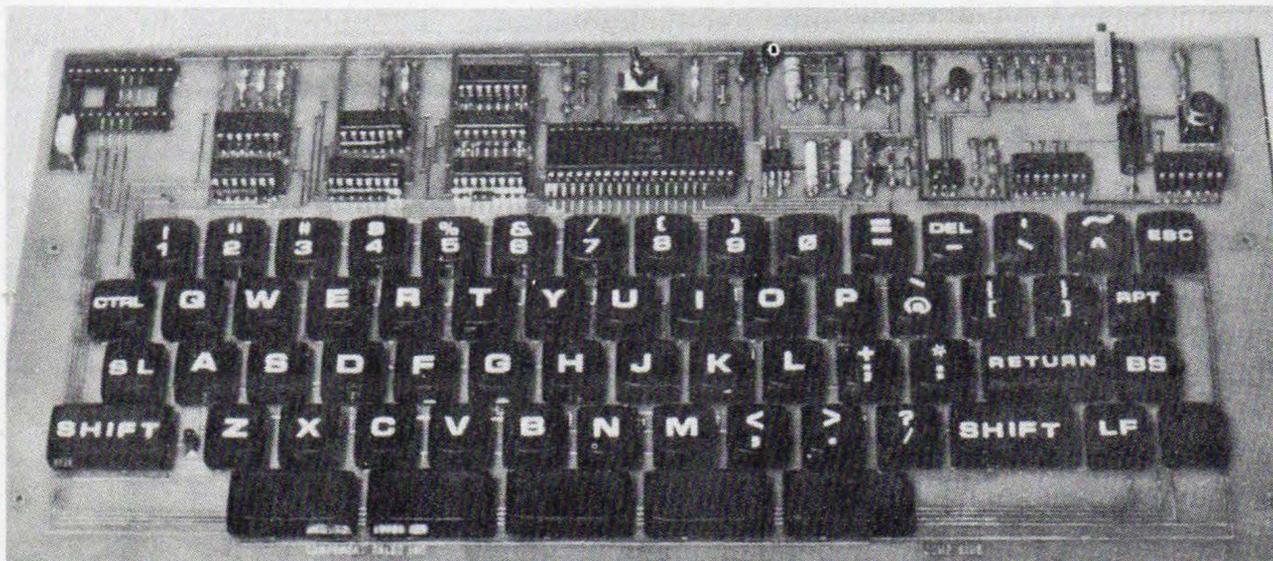
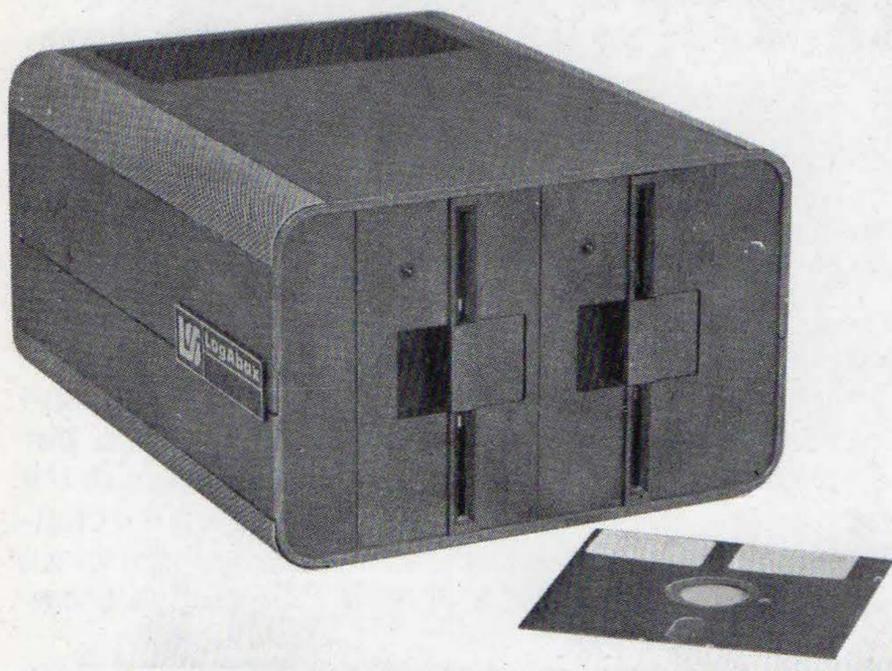



Figura 5 — Tastiera con logica di interfacciamento.

Figura 6 — Floppy-disk.

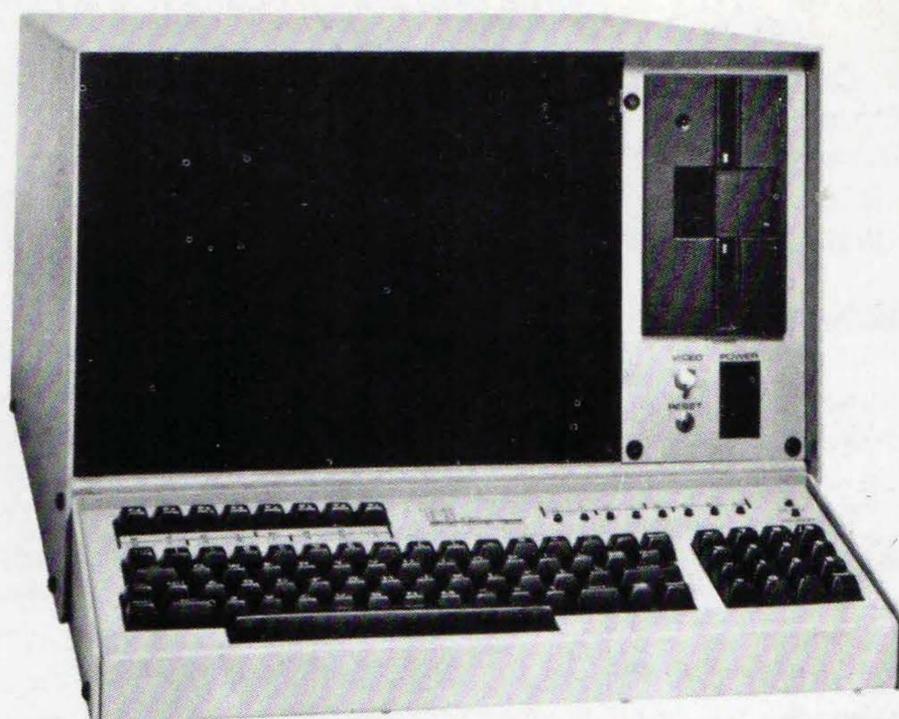
Figura 7 — Sistema microcomputer in configurazione integrata.

plicazione richiesta, necessitano copie fisiche delle elaborazioni del computer. Può essere utilizzata come alternativa al dispositivo di visualizzazione, ma è più costosa, più lenta e più rumorosa.

Memorie di massa: sono quei dispositivi, forniti di supporto magnetico, per la memorizzazione di dati e programmi, ossia di tutte quelle informazioni che, quando il computer è spento, non devono essere «dimenticate». Questa esigenza è evidente per dati acquisiti in tempi successivi e destinati ad essere archiviati. È comunque necessaria la presenza di dispositivi di tal genere anche per memorizzare i programmi, in quanto il corredo di software di un computer può essere notevole in funzione delle esigenze che deve coprire e che, in linea di principio, sono senza limite, soprattutto in ambito personal computer.

I tipi di memoria di massa più comuni sono:
- cassette magnetiche
- floppy-disk
- minifloppy.

Le **cassette magnetiche** sono i dispositivi meno costosi. Nei personal computers generalmente vengono utilizzate cassette audio per la loro ulteriore economicità rispetto a quelle digitali. Gli svantaggi di tali dispositivi sono: 1) l'attivazione della scrittura/lettura richiede l'intervento dell'utente; 2) sono estremamente lenti; 3) le informazioni sono memorizzate sequenzialmente, e quindi la ricerca dati/programmi è fatta nella stessa maniera; 4) offrono una non altissima affidabilità; 5) qualora si voglia memorizzare un notevole volume di informazioni, sono insufficienti.

Più professionali sono invece i **floppy-disk** (fig. 6) o i più recenti minifloppy. Il floppy-disk è ormai la memoria di massa standard per i microcomputers. Il supporto è costituito da un «disco» non rigido la cui superficie è trattata magneticamente. Il disco è mantenuto costantemente in rotazione da un motore; inoltre sulla superficie è posizionata una testina di lettura/scrittura che è in grado di muoversi radialmente. Una volta posizionata su una traccia, la testina legge/scrive sequenzialmente per tutto lo sviluppo della circonferenza della traccia.

Il floppy-disk è notevolmente più costoso della cassetta magnetica,ma offre i seguenti vantaggi: 1) maggiore capacità di memoria; 2) accesso alle informazioni molto più rapido; 3) acquisizione diretta dei dati/programmi; 4) gestione completamente automatica, ossia solo tramite computer.

Il computer

Un completo sistema microcomputer (fig. 7) è, pertanto, formato da:
- C.P.U.
- Memoria
- Logica di governo di ingresso/uscita
- Tastiera
- Video
- Cassetta/Floppy-disk

Questa configurazione può essere utilizzata come punto di riferimento, (anche se spesso è utile avere una stampante): le differenti variazioni rispetto a questa configurazione, in più o in meno, dipendono *dal tipo di utilizzo.*

Il software

È necessario, a questo punto, completare il discorso precedentemente accennato sul software.

Il software è generalmente diviso in:
- software di sistema
- software applicativo.

Il *software applicativo* è l'insieme dei programmi sviluppati per problemi specifici dell'utente.

Il *software di sistema* è quello fornito dal costruttore per rendere operabile il computer. È parzialmente residente su ROM, ma per la maggior parte è fornito su supporto magnetico.

Il software di sistema dipende dall'utilizzo che si fa del computer e quindi dal tipo di sistema scelto. Pertanto vengono qui semplicemente elencati i vari programmi che possono accompagnare il computer:

Editor. Questo programma aiuta l'utente nella preparazione/modifica dei programmi sorgenti facilitando le operazioni di aggiunta/cancellazione/modifica di linee di programma, etc.

Assembler. Questo programma traduce il linguaggio simbolico del programma scritto con *Editor* in linguaggio macchina.

GOULD biomation

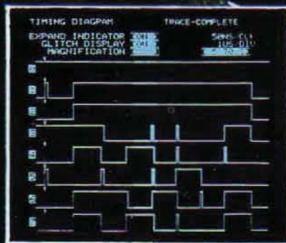
la più vasta gamma di logic analyzer

Il K100-D offre anche la rappresentazione dei dati Special Mode, molto utile per controllare l'Interface Bus IEEE 488.

Schermo incorporato con rappresentazione in funzione del tempo e dei dati.

Funzionamento asincrono fino a 100 MHz.

tastiera gestita da μP

1. Status Display

2. Timing Display

3. Data Display

1024 bits/can. più altrettanti per comparazioni

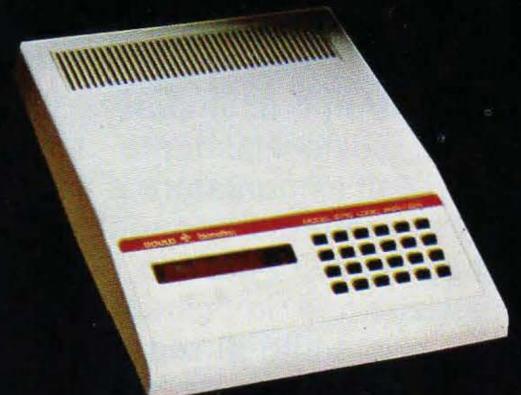
Sonde attive ad alta impedenza realizzate appositamente per il K100-D.

16 canali d'ingresso espandibili a 32 per risolvere anche particolari problemi connessi ai μP.

NUOVO Il Logic Analyzer K100-D è 5 volte più veloce e ha una memoria 4 volte maggiore del modello concorrente più direttamente comparabile. Non a caso questo prodotto è stato realizzato dalla Gould Biomation, la più avanzata costruttrice mondiale di tali apparecchiature. Il K100-D, con la sua possibilità di funzionamento asincrono fino a 100 MHz, i 16 canali di ingresso espandibili a 32, la memoria di 1024 bits per canale, il controllo da tastiera gestito da microprocessore, lo schermo incorporato, la rappresentazione in funzione del tempo e dei dati, il modo di funzionamento Latch selezionabile canale per canale e le sue esclusive sonde attive ad alta impedenza, si può senz'altro definire il più veloce, potente e versatile Logic Analyzer a 16 canali oggi esistente.

Modello	Freq. (MHz)	Latch (ns)	Canali	Memoria
920-D	20	10	9	256 x 9
851-D	50	5	8	512 x 8
1650-D	50	5	16	512 x 16
2710-D	10	–	27	64 x 27
8100-D	100	3	8	2048 x 8
K100-D	100	5	16 + 16	1024 x 16
9100-D	100	5	9	1024 x 9
8200-D	200	1	8	2048 x 8
168-D	10	–	1	256 x 25
DTO-1	Digital Testing Oscilloscope: 1. Logic Analyzer - 2. Storage Oscilloscope 3. Go/No-Go Comparator			

Il modello 920-D è il più economico Logic Analyzer a 20 MHz, 9 canali, 256 bits per canale, latch mode, trigger combinatoriale su tutti i canali ed è particolarmente adatto per il SERVICE dei circuiti digitali.

L. 1.850.000* - consegna pronta

NUOVO Il modello 2710-D è stato realizzato principalmente per l'analisi dei circuiti a microprocessori. Ha 27 canali di ingresso, 10 MHz, 64 bits per canale.

L. 1.850.000* - consegna pronta

una gamma completa di strumenti elettronici di misura

elettronucleonica s.p.a.

MILANO - Piazza De Angeli, 7 - tel. (02) 49.82.451
ROMA - Via G. Segato, 31 - tel. (06) 51.39.455

elettronucleonica S.p.A.

Desidero

- ☐ ricevere maggiori informazioni sul Logic Analyzer Gould Biomation modello ....
- ☐ avere una dimostrazione del Logic Analyzer Gould Biomation modello ....

Nome e Cognome ....

Ditta o Ente ....

Indirizzo ....

* Nov. 78 - Pag. alla consegna, IVA esclusa

Debugger. Facilita l'analisi degli errori di un programma caricato con il *Loader* fornendo all'utente la possibilità di visualizzare/modificare contenuti di memoria/registri, etc.

Loader. Carica il programma oggetto, cioè in linguaggio macchina, nella memoria RAM per l'esecuzione.

Compilatore. Traduce le istruzioni scritte *in linguaggio sorgente ad alto livello* in linguaggio macchina. Ad ogni istruzione simbolica ad alto livello corrispondono più istruzioni macchina.

Interprete. Traduce e *contemporaneamente* esegue un programma scritto in linguaggio ad alto livello. A differenza del compilatore, l'interprete *deve coesistere* sempre col programma sorgente in esecuzione.

Sistema operativo. È un programma che controlla l'esecuzione di tutti gli altri programmi con i quali coesiste, allo scopo di facilitare l'utente nell'utilizzo completo di tutte le risorse, hardware e software, del sistema.

Il software di sistema è la parte più importante del computer ed anche la meno «capita». Mentre l'hardware è diventato poco costoso, altrettanto non è accaduto per il software; occorre perciò valutare attentamente l'acquisto di un computer alla luce di questa considerazione. Attualmente sono presenti sul mercato vari tipi di sistemi. È essenziale individuare le proprie esigenze per una scelta opportuna: in ambito didattico, ad esempio, un sistema di minime prestazioni operante con linguaggio macchina è senz'altro ideale, ma per altri utilizzi è un insieme di componenti di scarsa utilità.

Che cosa scegliere?

A questo punto è opportuno fornire una classificazione indicativa dei vari tipi di sistemi esistenti, allo scopo di permettere un orientamento di massima soprattutto per chi è giustamente confuso dai numerosi prodotti apparsi recentemente.

Mainframe microcomputers

Sono sostanzialmente formati dalla scheda di C.P.U., dall'alimentatore e da uno chassis contenente un certo numero di connettori per l'inserimento di schede aggiuntive. Possono avere o no un pannello di visualizzazione/impostazione di indirizzi, celle di memoria, contenuto di registri, etc.

Fondamentalmente le alimentazioni e tutti i segnali caratteristici del microcomputer sono portati ad un bus comune allo scopo di poter espandere il sistema.

Il vantaggio di questo tipo di microcomputer è la sua versatilità: aggiungendo le schede opportune può essere configurato nel modo desiderato.

Le caratteristiche sono di tipo «general purpose»; da parte del costruttore originario nonchè da parte di costruttori indipendenti sono fornite varie schede di espansione. Inoltre il software di sistema è solitamente molto ricco e comprende sistema operativo, compilatori, assemblatore, etc.

Alcuni nomi sono:

ALTAIR 8800b della MITS (microprocessore 8080 e Bus S-100)
IMSAI 8080 della IMSAI (microprocessore 8080 e Bus S-100)
CROMENCO Z-2 della Cromenco (microprocessore Z-80 e Bus S-100)
CHILD Z della General Processor (microprocessore Z-80)

Personal-Business Computers

La caratteristica comune di questi sistemi è quella di essere pronti per l'utilizzo, cioè di essere orientati verso un'utenza per nulla interessata allo sviluppo hardware del sistema stesso; pertanto, da una parte sono generalmente dotati di un certo numero di periferiche integrate col sistema quali ad esempio tastiera, video, unità magnetiche, dall'altra sono supportati da vari linguaggi di programmazione ad alto livello.

Sistemi appartenenti a questa categoria sono:

Sol-20 della Processor Technology
APPLE II della Apple Computer
VDP 80 della IMSAI.

Microcomputers su singola scheda

In questi sistemi, su una singola scheda ci sono C.P.U., memoria ROM, RAM e porte di I/O per l'interfacciamento seriale e parallelo. L'utilizzo va dalle applicazioni industriali al campo hobby computers. Forniti senza alimentatori, questi sistemi sono un'ottima scelta per applicazioni specifiche in cui l'acquisizione della potenza del microcomputer avviene ad un costo inferiore a quello richiesto da un mainframe. Richiedono un'utenza capace di lavorare in hardware, nonchè fornita dei sistemi necessari per sviluppare il software.

Prodotti indicativi sono:

TI 990/100 M della Texas Instruments (microprocessore TMS 9900)
OEM-80E della Mostek (microprocessore Z-80).

Sistemi didattici

Progettati per l'istruzione in programmazione (linguaggi macchina) e nel progetto hardware, partono da una configurazione minima che solitamente può essere espansa con schede aggiuntive. Caratteristica comune è quella di essere forniti di unità di I/O (tastiera esadecimale e visualizzatori a LED) e di programmi di monitor più o meno completi. In tal modo è possibile il caricamento da tastiera di semplici programmi, il debug e l'esecuzione.

Accanto agli ormai vetusti KIM 1 e MMD1, sono indicativi di questo tipo di sistema:

NANOCOMPUTER NB Z-80 della SGS (microprogessore Z-80)
SYM-1 della SYNERTEK (microprocessore 6502)

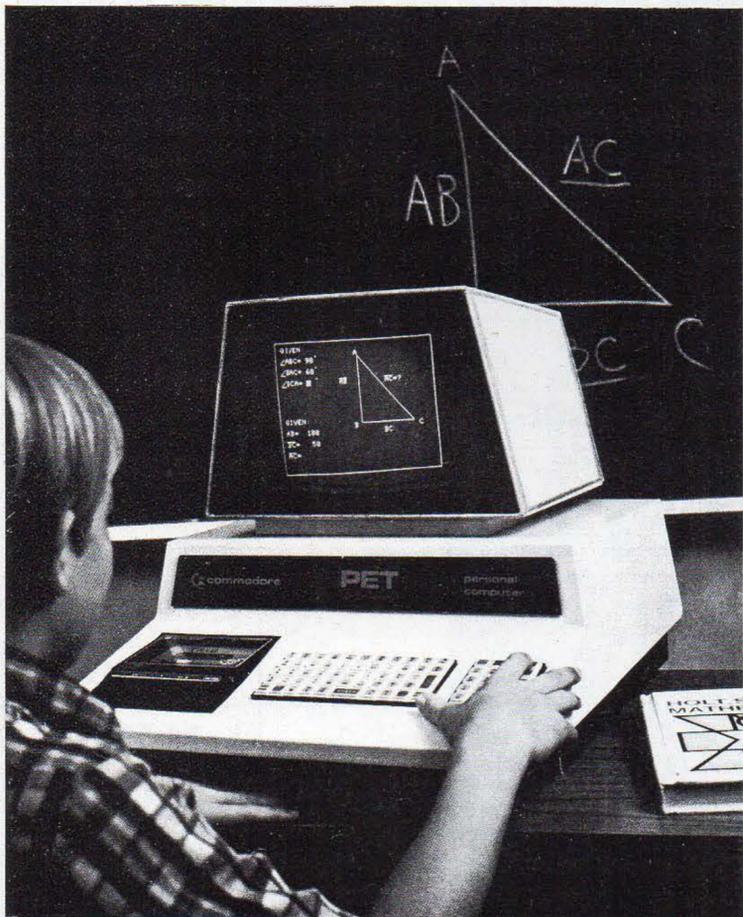
che permettono, tramite schede aggiuntive, di portare il sistema ad un livello più sofisticato. Dato il loro scopo, sono dotati di un'ottima documentazione.

Conclusione

Questa panoramica sugli elementi caratteristici di un microcomputer e sulle varie configurazioni offerte dal mercato ha il fine di chiarire il mondo del computer soprattutto a chi vi si accosta per la prima volta. Comunque un'indicazione valida per tutti è quella che, per utilizzare con piena efficacia le possibilità offerte da questi sistemi, occorre mettere a fuoco il più chiaramente possibile l'utilizzazione, e che le applicazioni di questi sistemi hanno come limite solo la nostra immaginazione. ■

HOMIC

Presenta in Italia i computer personali

COMMODORE PET E RADIO SCHACK TRS-80
I PERSONALI ALL'AVANGUARDIA

- Per la scuola
- Per il laboratorio
- Per il Club

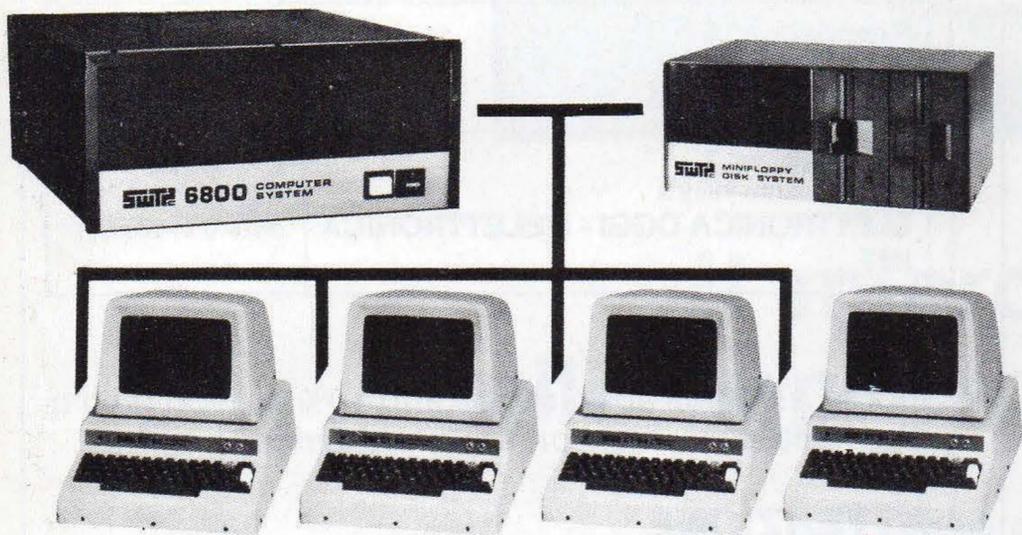
ELENCO DISTRIBUTORI HOMIC

concessionario per Roma:
MICRODATASISTEM
V.le Giulio Cesare, 199
Tel. (06) 314600 - ROMA

concessionario per la Liguria:
K-BYTES - di Alberto Capini & C.
Via Milazzo, 2
Tel. (010) 581709 - GENOVA

concessionario per Bergamo e Pavia:
INFOPASS S.A.S.
Via Trieste, 21
Tel. (02) 5271289 - SAN DONATO MIL.

concessionario per Latina:
LA CASA DEL COMPUTER
Via della Stazione, 9/15
Tel. (0773) 23585 - LATINA SCALO

SWTPC 6800 il potente microsistema operante in time-sharing

- Per la gestione di piccole-medie aziende
- Per la istruzione programmata nella scuola e nei laboratori linguistici
- Per lavori scientifici

NASCOM Z80 l'economico sistema in KIT operante in assembler e basic

per
- l'industria
- la scuola
- l'hobbystica

HOMIC

Bottega di dimostrazione: P.za de Angeli 3
Ufficio: via Dante, 9 Milano tel 809456

Il microcomputer nelle applicazioni gestionali *

di **Rodnay Zaks** - *Sybex*

Introduzione

Per la prima volta il progresso della tecnologia rende possibile utilizzare il computer nell'ambito delle piccole aziende per meno di cinquemila dollari. *Ma è realmente vero?* La risposta non è immediata, in quanto *pur essendo un sistema microcalcolatore in grado di fornire, realmente, prestazioni utili ad un'azienda, non è altrettanto vero che vi siano sistemi completamente validi.*

La carenza essenziale dei sistemi microcalcolatori attuali non è a livello *hardware,* ma a livello *software.* Questa è sempre stata la situazione, da quando i computers sono stati introdotti, e la cosa si è ripetuta ogni volta che è stata introdotta una nuova generazione di hardware: l'hardware necessario per realizzare un sistema per applicazioni aziendali può realmente essere acquistato per cifre comprese fra 5.000 e 20.000 dollari. Il problema è il software che comincia solo ora a diventare disponibile.

Naturalmente esistono molte possibilità in funzione delle prestazioni che si desiderano (vedi figure 1 e 2). Per scegliere in modo conveniente un sistema microcalcolatore, *è necessario comprendere* le possibilità delle varie soluzioni disponibili attualmente, poichè *non*

* Tratto da: Rodnay Zaks: «An introduction to personal and business computing». Sybex, Berkeley, Cal, 1978, di prossima pubblicazione in versione italiana presso la Jackson Italiana s.r.l.

Figura 1 — Il sistema VDP-40 della IMSAI. Il processore è l'8085. Il bus di sistema è l'S-100. Può essere utilizzato come terminale intelligente o come business computer. È supportato da IMDOS (IMSAI's Multi-Disk Operating System). Lavora in BASIC (quattro versioni) e in FORTRAN IV.

esiste «il meglio». Non c'è una scelta «general purpose» che si adatti a tutte le applicazioni.

Capire quale è l'hardware necessario e che cosa è disponibile è relativamente semplice. Il problema più complesso e difficile è capire le capacità software richieste. È su questo punto che la maggior parte degli interessati sbaglia nel comprare un sistema per applicazioni aziendali. Questi errori sono di solito più costosi di quelli fatti sull'hardware. Tipicamente gli investimenti nel software di un sistema diventano ben presto gli investimenti dominanti. Un computer inadeguato limiterà la possibilità di crescita del sistema medesimo: il passaggio ad un altro differente potrebbe essere costoso.

Per queste ragioni il lettore è vivamente incoraggiato a studiare e capire i concetti software come quelli hardware.

Applicazioni dei computers in campo aziendale

Ogni azienda ha bisogno innanzitutto di tenere un certo numero di archivi. Gli archivi più conosciuti sono: conti attivi, conti passivi, inventario, registro generale. Archivi supplementari possono essere: personale, lista dei clienti, lista degli indirizzi per l'invio di materiale pubblicitario, lista delle vendite, lista dei fornitori, situazione di cassa, beni dell'azienda, ed altro ancora.

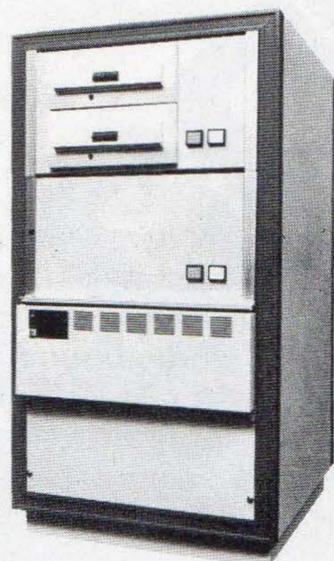



Figura 2 — Il sistema C3-B della Ohio Scientific. Ha tre processori: 6502A, 6800 e Z80. È fornito sia di floppy-disk che di disco. Può essere espanso per applicazioni multi-utente: in questo caso la memoria RAM può raggiungere i 768 Kbytes, in 16 partizioni.

La contabilità è tenuta di solito o da un contabile oppure tramite dispositivi elettromeccanici o computer o una combinazione di tutto questo.

Inoltre, per gestire gli archivi, un'azienda applica tecniche procedurali specifiche di ognuno di essi. Per esempio, un *programma per il libro paga* opererà sull'*archivio del personale* e genererà resoconti del libro paga, nonchè tabulati di controllo. Un *programma di imposta* opererà sui *resoconti di vendita* e sull'*archivio del personale* per produrre i *resoconti di imposta* richiesti. Un programma di *procedura di transazione* gestirà aggiornamenti di archivi specifici, il cambiamento o l'introduzione di nuovi dati. Un esempio tipico è una nuova vendita: il programma di transazione utilizzerà l'archivio inventario, l'archivio dei fornitori, l'archivio dei clienti, e forse altri. Poi aggiornerà e stamperà i rapporti.

Analogamente una *procedura di merci in arrivo* manipolerà le spedizioni in arrivo per introdurle nell'archivio inventario, controllerà l'esistenza di ordinazioni non evase, ed aggiornerà la lista dei *conti passivi.* Ogni pagamento ricevuto aggiornerà la lista dei *conti attivi* e la lista della *situazione di cassa.*

Oltre ai programmi principali, deve essere disponibile un certo numero di programmi supplementari, per poter avere resoconti utili.

È importante notare un principio abbastanza semplice:

1) Gli archivi devono essere creati e mantenuti.
2) Dovrebbero essere disponibili programmi per fare da interfaccia tra l'utente e gli archivi e per fornire le funzioni operative richieste.

Sfortunatamente, in un reale sistema per applicazioni aziendali questa è solo *una parte delle elaborazioni necessarie.* Infatti, nella maggior parte delle aziende, è abbastanza semplice il mantenimento di un *singolo* archivio; ma la maggior parte delle elaborazioni necessarie è data dalla esecuzione simultanea di *riferimenti incrociati* tra molteplici archivi e del loro *aggiornamento automatico.*

Facciamo un esempio. Viene ricevuta un'ordinazione per posta che è quindi elaborata dal programma *gestione delle transazioni.* La vendita viene introdotta nell'*archivio delle vendite* della giornata. Segue ora tutta una complessa serie di passaggi: come risultato, il nome del cliente verrà aggiunto automaticamente all'*archivio dei clienti.* Inoltre il suo nome verrà probabilmente codificato in funzione dell'acquisto che ha fatto, o dell'ammontare dell'acquisto, o della sua attività di lavoro. Poi, prima di cominciare ad elaborare l'ordinazione, si controllerà il credito del cliente. Se la vendita non è «vietata» dal programma *gestione dei crediti,* il prossimo passo è accettare l'ordinazione. Verrà poi verificata nell'archivio dell'*inventario dei prodotti da vendere* la disponibilità degli articoli ordinati. In questo esempio, vengono ordinati tre articoli: A, B e C. A e B sono in magazzino, C manca. Allora vengono generati una *fattura* per il cliente, una *bolla di uscita* ed un *ordine inevaso;* quest'ultimo è aggiunto alla relativa lista. Nel nostro esempio, l'articolo B è disponibile in magazzino, ma ne sono rimaste solo quattro unità. La *lista d'inventario* è strutturata con un campo speciale che specifica il livello delle riordinazioni. Il livello di riordinazione dell'articolo B è quattro. Il passo successivo di questa transazione sarà che un ordine di approvvigionamento sarà generato per l'articolo B per

Computer PET2001

IL PRIMO POTENTE COMPUTER PERSONALE

HARDEN S.p.A. DIVISIONE ELETTRONICA

26048 SOSPIRO (Cremona) Italia Tel.(0372) 63136 ric. aut. - Telex. 320588

Figura 3 — «Menu».

Figura 4 — Il «submenu» gestionale.

Figura 5 — Archivio conti attivi.

una quantità standard di 25 articoli (il numero 25 era specificato nell'archivio d'inventario). L'indirizzo del fornitore è ottenuto dall'*archivio dei fornitori.*

Questa semplice operazione di vendita ha richiesto l'uso di cinque archivi all'interno del nostro sistema e di vari programmi di elaborazione. In casi specifici, può essere anche necessario aggiornare, verificare o modificare archivi supplementari, o eseguire funzioni supplementari. Dovrebbe essere chiaro da questo esempio che, per essere veramente utile, un sistema gestionale deve poter accedere, modificare ed elaborare vari archivi.

Sfortunatamente, la maggior parte dei sistemi per applicazioni aziendali che utilizzano i microcomputers disponibili oggi sul mercato non eseguono un servizio completo. Infatti di solito forniscono la gestione di un *singolo archivio*, e non automatizzano completamente il processo di transazione. Molto deve essere fatto «a mano».

«Word processing»

«Word processing» si riferisce all'operazione di una macchina da scrivere computerizzata, dove l'utente può facilmente cambiare, modificare o dare forma al testo. Esso richiede un programma «editor», che è una prestazione standard sui computers tradizionali. Il costo del processore è diventato così basso, che può essere adibito ad una funzione quale il «word processing», cosicchè tali sistemi si stanno moltiplicando.

La maggior parte usa una Selectric o una macchina da scrivere analoga. Per contrasto, i sistemi per applicazioni commerciali offrono l'opzione di display e/o la configurazione multi-terminale.

Esempio di utilizzo.

Il tipo di programma e le nostre scelte sono precisati in risposta a domande che appaiono sullo schermo del terminale video.

Inizialmente, il sistema visualizzerà un «menu». Un «menu» è semplicemente una lista di domande (vedi figura 3). La domanda formulata dal computer è evidenziata con uno o più «caratteri di sollecito», che indicano che il microcomputer aspetta una risposta.

Supponiamo di aver digitato il numero 2 e quindi di aver scelto il programma «gestione». Il sistema lo carica automaticamente prelevandolo da disco. Appare allora un elenco di opzioni (vedi figura 4).

Specifichiamo «conti attivi». A questo punto, il sistema può richiedere che sia inserito un altro disco. Supponiamo di no. Sullo schermo comparirà un altro elenco (vedi figura 5).

Specifichiamo «nuova vendita»: il sistema richiederà a questo punto tutti i dati necessari per registrare la transazione.

Il dialogo col sistema è ora interattivo, almeno fino a quando la transazione non è completata. (In aggiunta il sistema dovrebbe verificare la validità dei dati in fase di introduzione; ossia che non vi siano errori grossolani). Poi è stampata la fattura e in seguito sono aggiornati i vari archivi interessati (conti attivi, inventario, lista dei clienti, etc.).

Software esistente

La maggior parte degli attuali microcomputers offre l'hardware necessario per eseguire tutte le operazioni che sono state descritte, a velocità sufficiente. Sfortunatamente, *non sono ancora stati sviluppati programmi per gestioni aziendali complete, che automatizzino tutte le mansioni richieste.*

Dalla descrizione precedente è evidente che la gestione degli archivi e i vari programmi di processing sono complessi: queste opportunità esistono solo per i computers più grandi.

Il costo per sviluppare tali programmi è molto maggiore del costo per sviluppare l'hardware; per questa ragione i costruttori non sono di solito ansiosi di svilupparli, in quanto fanno buoni affari vendendo hardware. Ditte di consulenza e software houses tradizionalmente coprono questa carenza vendendo programmi specializzati o package agli utenti di tali sistemi.

La maggior parte dei business computers attualmente disponibili offrono pochi programmi per gestioni aziendali; di solito forniscono un file system di caratteristiche generali, che permette all'utente di creare e manipolare gli archivi simbolicamente.

Questo è insufficiente. Devono esistere programmi specializzati di processing, che permettano la raccolta automatica dei dati, l'aggiornamento dei vari archivi comunque interessati, etc.

La maggior parte dei programmi disponibili risolve solo *uno* dei problemi. Di solito c'è un programma dei conti attivi, un programma *separato* dei conti passivi, un programma *separato* d'inventario e così via.

Questi «packages» separati sono utili per elaborare archivi che siano indipendenti l'uno dall'altro; *purchè il numero delle transazioni sia elevato per ognuno di essi, il servizio fornito è reale.* Comunque, essi fanno solo parte del lavoro. A carico dell'utente è richiamare i vari archivi che possono essere coinvolti in una singola transazione.

■

Fra le periferiche più prestigiose
eltron propone Data Media,
i supercollaudati video-terminali a grandi e medie prestazioni

ELITE 1521. Terminale basso costo TTY compatibile. Schermo da 1920 caratteri, tastiera separata da 87 tasti per un set di 128 caratteri, codice ASCII. Interfaccia EIA RS232C o 20 mA current loop. Velocità da 110 a 9600 bds, impaginamento in ROLL MODE con indirizzamento automatico X-Y del cursore, TAPE MODE, uscita RS232 per stampante, uscita segnale composito per monitors, costruzione a norme MIL.
ELITE 3025. Terminale bufferizzato per EDITING ad alta flessibilità. Funziona in PAGE e ROLL MODE con 2 K di memoria video. Possibilità di operare con formati protetti, reverse video, correzioni e cancellazioni automatiche, print page tramite uscita stampante, tabulazioni e formattazioni programmabili. Caratteristiche Hardware simili al mod. 1521.
ELITE 3045. Simile al 3025 ma con funzionamento sia in ASCII che in APL.
ELITE 3052. Terminale bufferizzato compatibile DEC VT 52®.
ELITE 4000. Terminale programmabile ad alta flessibilità costituito da un sistema a microprocessore con memoria espandibile fino a 32 K MOS RAM e 4 K ROM. CUP e linguaggio di programma 8080. Set di caratteri 254 full ASCII. Organizzazione display a blocchi programmabili con accesso diretto alla memoria.

Via Corsica, 14H - 25100 Brescia
Tel. (030) 55026 - 41554 - 44414 - Twx 30813

TRIBUNA

Il nuovo modo di vendere i calcolatori

di Sergio Focardi

Direttore Commerciale della Computeria

L'introduzione sul mercato dei Personal Computer ha aperto la possibilità di impiego dell'elaboratore a fasce di mercato finora non raggiungibili con problemi di costo, inaugurando nel contempo dei criteri di commercializzazione diversi da quelli tradizionali dell'informatica e più simili alla vendita di prodotti di largo consumo quali ad esempio le apparacchiature HI-FI.

L'elaboratore elettronico è stato per molti anni uno strumento completamente inaccessibile ai privati e alle piccole imprese. La complessità di una istallazione EDP e l'entità dell'investimento richiesto rendevano possibile l'impiego degli elaboratori solo a grandi imprese.

Gli elevati prezzi di vendita e di affitto dei calcolatori e i grandi margini di profitto conseguibili su ciascuna singola vendita imponevano una organizzazione di vendita estremamente aggressiva con costi di commercializzazione conseguenti particolarmente alti.

Ciascuna singola trattativa di vendita andava seguita con opportuna strategia e con grande dispendio di risorse umane e materiali, destinate a completare un'opera di convincimento accurata e capillare.

La prima svolta nella commercializzazione dei prodotti per la informatica si è avuta con l'avvento dei minicomputer.

L'azione di vendita, nel campo dei minicalcolatori, si affidava alla qualità e al prezzo del prodotto offerto piuttosto che all'azione commerciale in senso stretto.

Parimenti ridotta è stata, con i minicalcolatori, l'azione di «formazione dell'opinione» esercitata dalle case costruttrici che si sono limitate, per molti anni, ad offrire il loro prodotto lasciando che fosse il cliente a vederne la possibile applicazione.

L'evoluzione tecnica dei mini, e l'esigenza delle case costruttrici di espandere il loro mercato verso settori più profittevoli hanno fatto crescere verso l'alto l'applicabilità dei mini, divenuti ormai, per costo e dimensione, degli elaboratori medi.

Il mini non è più l'elaboratore per tutti, ma è piuttosto un mezzo per realizzare, con un buon rapporto prezzo/prestazioni, applicazioni sofisticate.

Significativo è in questo senso l'atteggiamento dei più grossi produttori di mini che tendono ormai a concentrare la loro azione di vendita diretta su tratta-

tive ad elevato profitto marginale che possono remunerare i costi di organizzazioni di vendita divenute ormai considerevolmente costose.
L'introduzione dei microcalcolatori e dei Personal Computer sembra viceversa poter veramente realizzare le possibilità di una informatica alla portata di tutte le applicazioni sia per i costi, che per la facilità di impiego dei mezzi e per le tecniche di commercializzazione.

È interessante sottolineare i punti più significativi di questa nuova svolta della tecnologia e del mercato dell'informatica.
L'esperienza dominante in questo senso è quella statunitense, che la UNI-COMP ha riportato oggi in Italia con l'apertura della Computeria, centro di vendita dei Personal Computer.
Il mercato dei Personal Computer è esploso da solo, senza essere stato preparato da estese azioni di vendita. L'esigenza di prodotti a costi realmente bassi e con caratteristiche di operatività alla portata di tutti era talmente forte che non appena tali prodotti sono divenuti disponibili la richiesta ha immediatamente superato l'offerta.
Si nota, in conseguenza, nel mondo dei Personal Computer, un differente modo di concepire la vendita.
Negli Stati Uniti si è assistito all'apertura dei negozi di calcolatori, veri punti di vendita al dettaglio.
È avvenuta una completa inversione di tendenza commerciale: qui è il cliente che va liberamente al fornitore, spinto da una esigenza che si impone da sè, anzichè essere il fornitore che cerca il cliente.
Il cliente chiede di poter esaminare, in modo pratico, differenti alternative per trovare i prodotti che meglio si adattano alla sua applicazione.
Questo impone che il punto di vendita sia svincolato da un particolare fornitore.
In effetti le maggiori iniziative in questo settore puntano su un ampio catalogo di prodotti e offrono in vendita al cliente il meglio della produzione, consigliandolo, con serietà e competenza, verso una scelta che corrisponde alle sue reali esigenze.
Lo svincolamento dal fornitore è reso possibile dalla componibilità dei Personal Computer i quali possono essere configurati «su misura» usando componenti di differenti produttori. Importantissimo, in questo senso, il concetto del Bus S-100 un bus divenuto ormai uno standard industriale che consente di comporre ed espandere ogni sistema che rispetti questo standard usando periferiche ed interfacce prodotte, oggi, da più di 100 differenti case.
Questa apertura costituisce una garanzia di serietà in quanto il cliente non viene mai forzato a scegliere una soluzione particolare, ma può attingere ad una varietà di prodotti tra cui, nella maggioranza dei casi, potrà trovare ciò che veramente fa per lui salvaguardando il suo investimento con la garanzia di espandibilità futura.
Tutto ciò richiede ovviamente un adeguato supporto. Supporto tecnico per garantire l'assistenza al prodotto, quando esso diventa operativo, ma anche supporto di marketing per fornire al mercato il necessario livello di informazione tecnica e commerciale.
È nostra opinione che una seria attività di supporto tecnico e commerciale sia una delle chiavi di volta del successo di questo mercato così nuovo e così promettente.
Le enormi possibilità tecniche e applicative che si aprono con questi prodotti a basso costo esigono, da parte degli operatori del settore, una grande chiarezza di idee e di programmi, sia per la pura vendita che per gli aspetti collaterali di formazione e supporto.
Con questa visione del mercato abbiamo costituito la Computeria.
La Computeria si rivolge sia al privato che intende acquistare un singolo sistema per uso hobbystico o professionale, sia alle organizzazioni che intendono distribuire i prodotti importati dalla Computeria o realizzare, a partire da essi, un loro prodotto finito.
Al privato la Computeria propone il suo centro di dimostrazione e di vendita a Cinisello Balsamo, in cui può esaminare i prodotti offerti e, consigliato da personale esperto, scegliere su un catalogo che comprende più di 100 fornitori.
Ai distributori e agli O.E.M. la Computeria propone un discorso commerciale basato su una reale comprensione della problematica di questo nuovo mercato e su un'offerta di prodotti e servizi che ci sembra estremamente interessante per chi intende operare in questo nuovo settore seriamente e con profitto.

Progetto di una unità a cassetta magnetica

di **Robert Suding** - *Digital Group Inc.*

Una accurata analisi dei limiti e delle possibilità di un supporto magnetico a cassetta ha permesso di sviluppare un sistema sicuramente interessante per le ottime prestazioni e la notevole semplicità.

Le unità a cassette magnetiche presentano spesso problemi applicativi a causa della difficoltà riscontrata nell'ottenere contemporaneamente una elevata velocità operativa e una sufficiente affidabilità.

Molte unità a cassetta offrono infatti prestazioni deludenti che costringono l'utente a lunghe attese o a noiose ripetizioni per ottenere con successo il trasferimento di dati o programmi.

I limiti di questi sistemi sono solitamente da imputarsi all'impiego di registratori a cassetta di scarsa qualità o di componenti di recupero, insieme ad una messa a punto poco accurata dei circuiti ad audiofrequenza di modulazione e demodulazione del segnale di supporto dei dati.

L'interfaccia per unità a cassette, presentata in questo articolo, consente di realizzare un sistema particolarmente affidabile e soprattutto veloce: offre infatti la possibilità di operare a 1100 Baud (lo schema ha funzionato regolarmente fino a 1750 Baud, dove insorgono i primi fenomeni critici), con una sensibile riduzione dei tempi di trasferimento in rapporto alle usuali cassette magnetiche a 300 Baud o ad altri supporti più lenti.

Il trasferimento di 24 Kbytes, ad esempio, richiede i seguenti tempi:

40' e 58" per un lettore di banda perforata a 110 Baud;
15' e 1" per una cassetta magnetica a 300 Baud;
4' e 6" per il sistema in oggetto operante a 1100 Baud.

Condizioni indispensabili per ottenere questi risultati sono l'impiego di un buon registratore a cassette con velocità del nastro accuratamente stabilizzata e l'utilizzo di ottimi nastri magnetici nonchè di componenti elettronici di buona qualità.

Prima di analizzare in dettaglio lo schema proposto, è bene soffermarsi su due importanti fattori limitativi della velocità di trasferimento, spesso sottovalutati in molte applicazioni:

1) come è noto l'informazione è memorizzata sul supporto magnetico modulando in frequenza una portante, secondo il valore binario dei dati digitali serializzati: l'impiego di una portante con frequenza troppo bassa non consente elevate velocità di modulazione e quindi alte Baud rate.
2) I due valori di frequenza associati ai valori binari memorizzati non devono essere correlati armonicamente, ossia la frequenza superiore non deve essere un multiplo esatto di quella inferiore. Questa situazione crea infatti dei problemi di discriminazione del segnale riprodotto, soprattutto nei sistemi che usano tecniche di rilevazione basate su filtri o retroazioni del tipo phase lock.

Il sistema presentato in questo articolo è stato dimensionato in modo da evitare le suddette limitazioni: impiega le frequenze, non correlate armonicamente, di 2125 Hz (mark) e 2975 (space). La seconda armonica della frequenza inferiore è a 4250 Hz, sufficientemente oltre la frequenza superiore di 2975 Hz, per non creare problemi di ambiguità nel rilevatore passabanda di quest'ultima.

Le frequenze impiegate sono più alte di quelle solitamente adottate nelle comuni cassette magnetiche (1200 e 2400 Hz), e ciò consente di elevare la velocità di trasferimento a 1100 Baud. Inoltre tra i valori scelti esiste un sufficiente margine per consentire una corretta discriminazione anche in presenza di fluttuazioni di frequenza dovute a variazioni della velocità del nastro.

Principio di funzionamento

Lo schema circuitale del sistema a cassette proposto è mostrato nelle figure 1 e 2.

Il circuito richiede per la sua alimentazione le tensioni di +5 e ±12 Volt.

La sezione «ricevitore» discrimina i due valori della portante di 2125 Hz e 2975 Hz, producendo in uscita un segnale logico «1» o «0».

Un amplificatore operazionale 741, indicato IC34, funziona come limitatore di ampiezza del segnale e impedisce che variazioni di ampiezza, dovute al riproduttore, possano influenzare il processo di discriminazione.

L'uscita del limitatore fornisce un segnale di circa 6 Volt pp, avente l'andamento di un'onda quadra con i fronti arrotondati secondo la sinusoide della frequenza di ingresso, costante in ampiezza indipendentemente dal volume del registratore.

Due filtri attivi passabanda (IC35), sintonizzati sulle due frequenze di 2975 Hz e 2125 Hz, amplificano cinque volte la propria frequenza di accordo. Il guadagno di ogni filtro è tanto minore quanto più la frequenza di ingresso si discosta da quella di accordo.

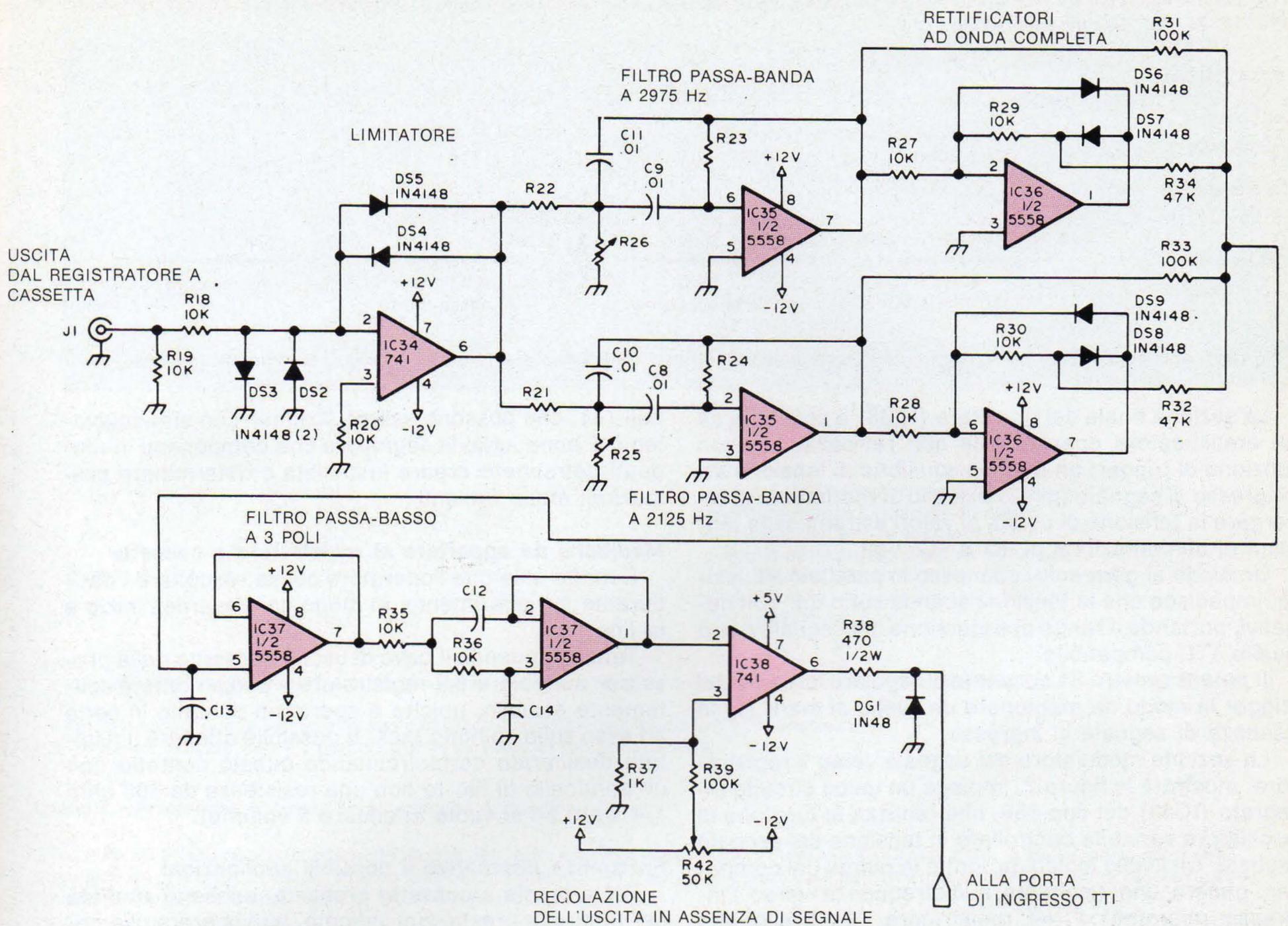



Figura 1 — Schema elettrico della sezione di ingresso dell'interfaccia per unità a cassette presentata dalla Digital. Uno stadio amplificatore e limitatore, costituito dall'operazionale IC34, riceve il segnale proveniente dal riproduttore a cassette; un secondo stadio, costituito da due filtri passabanda, seguiti da due rettificatori ad onda completa (IC36), discrimina le due frequenze supporto dei dati; infine un filtro attivo passabasso del terzo ordine, produce un segnale continuo a livello TTL, che può essere inviato direttamente ad un ingresso seriale del computer.

Il guadagno, l'ampiezza di banda e la frequenza di accordo sono fissati dalla rete costituita da tre resistori e due condensatori per ogni filtro. In particolare, le resistenze variabili R25 ed R26 consentono di aggiustare leggermente la frequenza di accordo dei filtri, compensando eventuali scostamenti dei valori teorici a causa delle tolleranze dei componenti.

Due rettificatori attivi ad onda completa producono impulsi rettificati sull'ingresso sommatore di IC37 (pin 5).

La frequenza superiore produce impulsi rettificati positivi, mentre quella inferiore impulsi negativi: le frequenze comprese tra 2125 Hz e 2975 Hz producono impulsi rettificati di minore ampiezza e segno differente a seconda della prossimità all'uno o all'altro estremo. Approssimativamente la frequenza centrale di 2550 Hz produce una uscita nulla al nodo sommatore di IC37.

Un filtro attivo passabasso a tre poli elimina qualsiasi traccia di pulsazione dovuta alla portante, senza influenzare la pulsazione più lenta di modulazione, associata all'informazione trasferita.

Se si utilizzano frequenze di trasferimento dei dati inferiori a quella proposta di 1100 Baud, è possibile migliorare il rapporto segnale-rumore moltiplicando i valori dei condensatori C12, C13 e C14 per l'inverso del rapporto tra le due Baud rate.

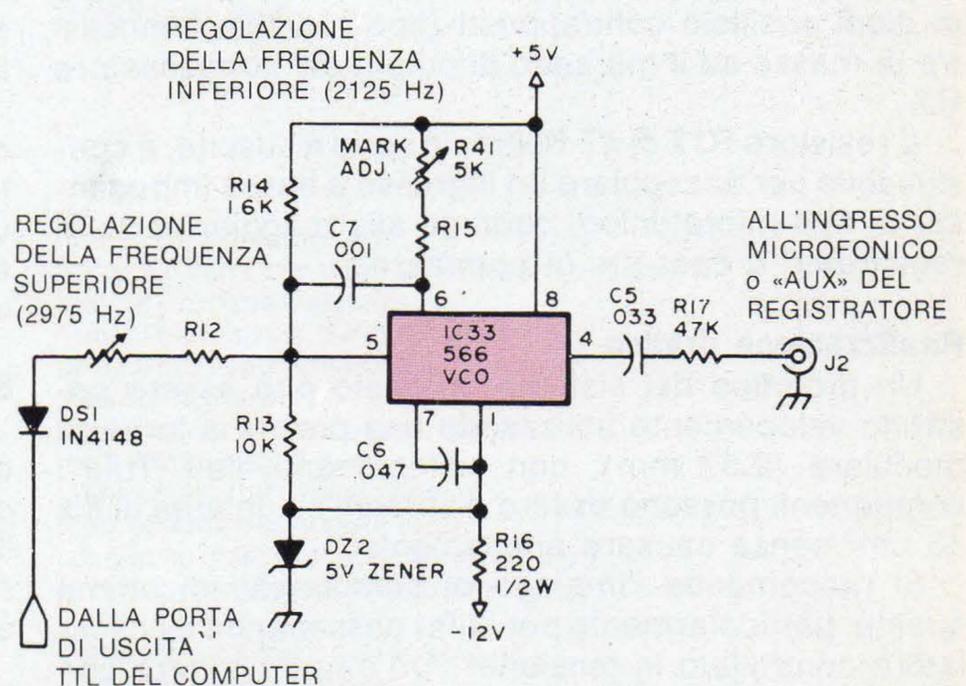



Figura 2 — Schema elettrico della sezione di uscita dell'interfaccia per unità a cassette. È costituita, molto semplicemente, da un oscillatore ad audiofrequenza, controllato in tensione direttamente dal segnale TTL associato ai dati. Questi possono essere serializzati su una porta di output del computer, impiegando un software che simula le funzioni di un UART.

Tabella 1 — Valori teorici dei componenti per un dimensionamento del sistema secondo alcune possibilità alternative. I valori si intendono inseriti negli schemi delle figure 1 e 2.

	R21	R24	R25	R22	R23	R26	C13	C12	C14	R12	R15
2125-2975Hz 1100 Baud	6.8K	68K	938*	4.7K	47K	697	.0056μF	.01	.015	2.7K	1.3K
1200-2400Hz 300 Baud	6.8K	68K	4173*	4.7K	47K	1162*	.0056μF	.01	.015	470K	2.7K
2125-2295Hz 100 Baud	6.8K	68K	938*	4.7K	47K	1301*	.0056μF	.01	.015	47K	2.7K
	filtro tono inferiore			filtro tono superiore			filtro passa-basso			modulatore	

* I valori indicati si riferiscono ai componenti teoricamente richiesti. Chiaramente si farà uso dei valori commerciali più prossimi.

La sezione finale del ricevitore (IC38) è costituita da un amplificatore operazionale non retroazionato con funzione di trigger; un minimo squilibrio di tensione tra l'ingresso di segnale (pin 2) e quello di riferimento fa divergere la tensione di uscita ai valori estremi delle tensioni di alimentazione di +5 e -12 Volt.

Un diodo al germanio, connesso in parallelo all'uscita, impedisce che la tensione scenda sotto 0.2 Volt negativi, portando il range di escursione del segnale entro quello TTL compatibile.

Il potenziometro P1 consente di regolare la soglia del trigger in modo da mantenere un livello di mark (1) in assenza di segnale di ingresso.

La sezione modulatore del segnale verso il registratore, mostrata in figura 2, impiega un unico circuito integrato (IC33) del tipo 556, che realizza la funzione di oscillatore variabile controllato in tensione dal segnale digitale. Un livello logico, prodotto in output dal computer, genera una risultante audiofrequenza verso l'ingresso microfonico del registratore. Un livello alto («1») causa l'emissione di un segnale a 2125 Hz, mentre un livello basso («0») a 2975 Hz.

Il segnale prodotto in uscita dal modulatore è triangolare simmetrico; è possibile ottenere una forma d'onda più prossima alla sinusoide, utilizzando una coppia di diodi al silicio contrapposti (tipo 1N914), connessi tra la massa ed il morsetto di output del condensatore C5.

Il resistore R17 di 47 Kohm, in serie all'uscita, è consigliabile per accoppiare un ingresso a bassa impedenza, di tipo microfonico, comune alla maggior parte di registratori a cassette in commercio.

Realizzazione pratica

Un prototipo del sistema proposto può essere costruito velocemente utilizzando una piastra a foratura modulare (2.57 mm), con collegamenti filati. Tutti i componenti possono essere distribuiti su un'area di 8 x 13 cm^2 senza causare affollamento.

Si raccomanda l'impiego di componenti di ottima qualità, particolarmente per i filtri passabanda e l'oscillatore controllato in tensione. Può infatti capitare che alcuni 556 di recupero presentino una scarsa stabilità di frequenza al variare della temperatura.

Il lay-out dei componenti sulla piastra non è critico, tuttavia è bene limitare la lunghezza dei collegamenti particolarmente nel limitatore, nei filtri passabanda e nello squadratore, al fine di evitare retroazioni indesiderate.

Non è vincolante l'impiego di amplificatori operazionali 741, che possono essere sostituiti con altri equivalenti. È bene tuttavia segnalare che componenti inadeguati potrebbero creare instabilità o determinare prestazioni meno brillanti.

Modifiche da apportare al registratore a cassette

È molto utile che l'operatore possa «ascoltare i dati» durante il trasferimento, in modo da rilevarne l'inizio e la fine.

Tuttavia, quando il cavo di uscita è inserito nella presa per auricolare del registratore, l'altoparlante è solitamente escluso, poichè è aperto un contatto in serie ad esso sullo spinotto jack. È possibile ottenere il risultato desiderato cortocircuitando questo contatto con un ponticello di filo (o con una resistenza da 100 ohm 1/4 watt, se si vuole attenuare il volume).

Frequenze alternative e possibili applicazioni

L'interfaccia a cassette proposta, sebbene studiata per fornire le prestazioni indicate, può operare ove necessario a diverse Baud rate o impiegare diverse frequenze di modulazione. La tabella 1 fornisce i valori dei componenti per alcune possibilità alternative.

Una possibile applicazione dello schema proposto, oltre a quella implicita di un sistema di memoria di massa a cassette, è la realizzazione di un sistema radiotelescrivente (RTTY).

I sistemi RTTY generalmente utilizzano le frequenze di 2125 Hz e 2295 Hz quali supporto dei dati. È possibile accordare i filtri passabanda su queste frequenze e utilizzare lo schema per interfacciare un ricevitore ad onde corte con un terminale dotato di interfaccia seriale.

Software

Il listing 1 è un esempio di gestione software, mediante un microprocessore 8080, della serializzazione dei dati verso l'interfaccia del sistema a cassetta. Le istruzioni marcate con il simbolo «*», contengono le costanti di temporizzazione riferite ad un clock di sistema di 2 MHz.

Chiaramente un componente tipo UART potrebbe essere usato in alternativa al software proposto, tuttavia la gestione software, pur appesantendo il lavoro del microprocessore, offre i vantaggi di una maggiore semplicità, minore costo e maggiore flessibilità. Si può infatti pensare di modificare dinamicamente la frequenza di trasferimento, campionando la durata degli stop-bit, al fine di compensare fluttuazioni di velocità del nastro.

Listing 1 - Programma di input da cassetta magnetica

La routine CASTOMEM trasferisce un blocco di dati dal supporto magnetico alla zona di memoria compresa tra le posizioni XXXX 16e YY00 16. Si suppone che i dati siano stati trasferiti in precedenza su cassetta mediante la routine MEMTOCAS (vedi listing 2). Si possono apportare delle modifiche alla routine proposta in modo da introdurla nel monitor di sistema.

```
      CASTOMEM   LXI   H,XXXXH      ;indirizzo iniziale del trasferimento
      STARTBYT   LXI   D,0008H      ;annulla D, poni E=8 (numero di bit)
      SYNCHLOO   IN    1            ;leggi dalla porta di input
                 ANI   1            ;evidenzia il bit 0
                 JNZ   SYNCHLOO     ;se diverso da zero attendi lo start bit
*2               MVI   B,0C0H       ;se uguale a zero (start bit) attendi fino
      WSYNCH     DCR   B            ;alla metà del primo bit dati
                 JNZ   WSYNCH
      GETDATA    IN    1            ;leggi dalla porta di input
                 ANI   1            ;evidenzia il bit 0
                 ADD   D            ;somma il nuovo bit ai precedenti
                 RRC                ;ruota la configurazione ottenuta
                 MOV   D,A          ;salva in D la nuova configurazione
*1               MVI   B,80H        ;ritardo tra due bit dati
      WDATA      DCR   B            ;attendi il nuovo bit dati
                 JNZ   WDATA
                 DCR   E            ;decrementa il contatore di bit
                 JNZ   GETDATA      ;se non è zero carica un nuovo bit
                 MOV   M,D          ;altrimenti memorizza il byte completo
                 INX   H            ;punta alla successiva posizione di memoria
                 MOV   A,H          ;prendi la parte più significativa dell'indirizzo
      $          CPI   YYH          ;confronta con l'indirizzo finale del trasferimento
                 JNZ   STARTBYT     ;se non ho finito attendo un nuovo start bit
                 HLT                ;fine trasferimento
```

Note:
- si assume che l'ingresso dati sia costituito dal bit meno significativo della porta 1 (direttamente connesso all'uscita del demodulatore).
- il caricamento procede sequenzialmente dall'indirizzo XXXX 16all'indirizzo YY00 16 escluso.
- le posizioni indicate con (*) contengono le costanti di temporizzazione.

Listing 2 - Programma di output verso cassetta magnetica

La routine MEMTOCAS trasferisce un blocco di dati, compreso tra le posizioni di memoria XXXX 16 e YY00 16 sul supporto magnetico, preceduto da una testata costituita da un tono basso continuo per la durata di 5 secondi. I dati inviati su cassetta possono essere ricaricati in memoria mediante la routine CASTOMEM (listing 1).

```
      MEMTOCAS   LXI   H,XXXXH      ;indirizzo iniziale del trasferimento
                 MVI   A,1          ;poni un livello alto
                 OUT   1            ;sul bit dati della porta di output
                 MVI   D,0AH        ;costanti di ritardo
      LEADER5S   MVI   B,0FFH       ;per generare una testata
      LEADER5X   MVI   C,0FFH       ;della durata di 5 secondi
      DEADER5Y   DCR   C            ;precedente i dati
                 JNZ   LEADER5Y
                 DCR   B
                 JNZ   LEADER5X
                 DCR   D
                 JNZ   LEADER5S
; a questo punto una testata di 5 secondi di tono basso è stata inviata
      BYTEOUT    MVI   C,9H         ;presetta il contatore di bit = 9
                 XRA   A            ;azzera il carry che deve generare lo start bit
                 MOV   A,M          ;carica il byte da trasferire
                 RAL                ;ruota i bit (carry = 0 è lo start bit)
      WNEXBIT    OUT   1            ;invia il bit meno significativo
*1               MVI   B,80H        ;ritardo tra due bit
      WOUTLOOP   DCR   B            ;decrementa il contatore di ritardo
                 JNZ   WOUTLOOP     ;se il tempo è scaduto
                 RAR                ;poni il nuovo bit in posizione
                 DCR   C            ;decrementa il contatore di bit
                 JNZ   WNEXBIT      ;se non ho finito procedo con un altro bit
                 MVI   A,1          ;output per lo stop bit (livello alto)
                 OUT   1
*3               MVI   B,0FFH       ;tempo di persistenza dello stop bit
      WIBDELAY   DCR   B            ;decrementa il contatore
                 JNZ   WIBDELAY     ;se il tempo non è scaduto attendi
                 INX   H            ;punta ai successivi byte in memoria
                 MOV   A,H          ;prendo la parte più significativa dell'indirizzo
      $          CPI   YYH          ;se non ho finito il trasferimento
                 JNZ   BYTEOUT      ;serializzo un nuovo byte
                 HLT                ;fine trasferimento
```

Note:
- si assume che l'uscita sia costituita dal bit 0 della porta di output 1 (connesso all'ingresso del modulatore)

Note di taratura

Per ottenere le migliori prestazioni dello schema proposto, si dovrà procedere ad una accurata taratura secondo la procedura elencata nel seguito.

1. Collegare un generatore sinusoidale ad audiofrequenza tra l'ingresso del limitatore e la massa. Si raccomanda l'impiego, ove possibile, di un generatore ad impostazione digitale della frequenza in uscita, in grado di coprire la banda tra 2 ÷ 3 KHz, con una ampiezza di segnale di 0.5 volt.

2. Applicare al circuito le tensioni di alimentazione di +5 e ±12 volt. Con un oscilloscopio osservare il segnale in uscita dal limitatore (pin 6 di IC34); il segnale dovrà apparire un onda quadra a fronti arrotondati, con ampiezza di circa 0.6 volt.

3. Porre il generatore sulla frequenza di 2125 Hz. Misurare l'uscita del filtro passabanda inferiore (IC35 pin 1) e regolare il trimmer R25 in modo da ottenere il massimo segnale in uscita.

4. In modo analogo porre il generatore sulla frequenza di 2975 Hz e regolare il trimmer R26 per il massimo segnale in uscita dal filtro passabanda superiore (IC35 pin 7). In entrambi i casi 3. e 4., a regolazione ultimata, variare leggermente la frequenza del generatore in modo da verificare che effettivamente il segnale in uscita dai filtri abbia la massima ampiezza in corrispondenza della frequenza di accordo.

5. Misurare la tensione al pin 5 di IC37 in funzione della frequenza di ingresso. In corrispondenza del tono inferiore (2125 Hz) la tensione deve essere negativa, mentre in corrispondenza di quello superiore (2975 Hz), positiva. Un mancato funzionamento a questo livello potrebbe essere causato da diodi difettosi o erroneamente invertiti o da cortocircuiti tra linee adiacenti.

6. Misurare la tensione all'uscita del trigger (catodo del diodo G1): variazioni di frequenza tra i due toni di accordo dovrebbero causare bruschi salti di tensione tra +5 e -0.3 volt.

7. Rimuovere il generatore ad audiofrequenza e cortocircuitare momentaneamente a massa l'ingresso J1. La tensione al pin 6 di IC34 dovrebbe essere stabile senza alcuna oscillazione. In queste condizioni portare il potenziometro di bilanciamento R42 in posizione tale da riscontrare in uscita una tensione negativa (catodo di G1). Successivamente regolare lentamente lo stesso potenziometro fino al punto in cui si verifica la transizione positiva della tensione.

8. Collegare nuovamente l'ingresso J1 al generatore sinusoidale e ripetere la prova indicata al punto 6. La soglia di frequenza intorno alla quale avviene la commutazione dovrebbe essere prossima a 2550 Hz. A questo punto, se le prove hanno dato esito positivo, la sezione demodulatrice dell'interfaccia è pronta ad accettare dati.

9. Collegare l'ingresso TTL del modulatore alla tensione di +5 volt, ed il connettore di uscita ad un frequenzimetro digitale. Regolare il potenziometro R41 fino ad ottenere una frequenza di uscita di 2125 Hz.

10. Collegare a massa l'ingresso del modulatore e regolare il potenziometro R40 fino ad ottenere una frequenza di uscita di 2975 Hz.

11. Raffinare le regolazioni indicate ai punti 9. e 10., sostituendo ai riferimenti di +5 volt e massa gli effettivi livelli TTL «1» e «0». Per ottenere questa condizione è possibile collegare il modulatore alla porta di output del microprocessore e programmare quest'ultimo affinché produca in uscita i livelli desiderati. L'interfaccia è ora tarata e pronta per il funzionamento.

Problemi realizzativi

Una precisa conoscenza dei problemi e dei punti critici che un certo sistema può causare, è spesso il primo passo verso la loro soluzione.

Nel seguito sono elencate sei classi di problemi inerenti alla realizzazione dello schema proposto.

1) *Registratore a cassette di scarsa qualità.* Come già accennato, le migliori prestazioni del sistema dipendono dall'impiego di un ottimo registratore a cassette, con velocità del nastro accuratamente stabilizzata.

2) *Problemi connessi al microprocessore.* Nell'articolo si propone l'implementazione di un software per la serializzazione dei dati digitali. Può accadere che questa soluzione sia inadeguata a talune organizzazioni del software, e quindi si renda necessario l'impiego di un UART.

3) *Problemi di collegamento.* In molti registratori è possibile invertire la posizione dei cavi di lettura e scrittura, con evidente mancato funzionamento.

4) *Problemi di accordo.* Un funzionamento poco affidabile può essere causato dai filtri attivi passa - banda, non perfettamente sintonizzati sui due toni della portante. Particolare cura deve essere rivolta alla procedura di taratura indicata.

5) *Cassetta magnetica difettosa.* Spesso può accadere che una cassetta magnetica, solitamente funzionante, generi improvvisamente errori nel trasferimento dei dati a causa di un danno dello strato di ossido del nastro. Per limitare questa eventualità, è bene introdurre e rimuovere la cassetta dal registratore con il nastro completamente riavvolto, per evitare abrasioni a zone significative di nastro durante questa operazione.

6) *Problemi circuitali e possibili cause.*

a) *segnale di ampiezza anormale in uscita dal limitatore:*
- nessuna uscita: verificare la presenza delle tensioni di ±12v su IC34
- livello troppo alto: diodi DS4 o DS5 invertiti o aperti.

b) *filtri passa - banda non funzionanti:*
- accordati su una errata frequenza (componenti fuori tolleranza)
- operazionali 5558 difettosi
- possibili cortocircuiti sulla rete di accordo

c) *rettificatori ad onda intera non funzionanti:*
- diodi aperti, invertiti o cortocircuitati.
- operazionale IC36 difettoso.

d) *filtro passa - basso difettoso:*
- componenti fuori tolleranza o possibili corticircuiti.
- operazionale IC37 difettoso.

e) *il trigger non produce livelli TTL:*
- diodo al germanio G1 invertito o aperto.
- eccessivo carico all'uscita (superiore ad un carico TTL).

f) *modulatore non funzionante:*
- integrato 556 difettoso.
- condensatore C6 in cortocircuito.

g) *oscillazione parassita ad alta frequenza del modulatore:*
- C7 non connesso.
- integrato 556 difettoso.
- C6 aperto.

h) *il modulatore non si accorda sui toni richiesti o non è stabile in frequenza:*
- C6 fuori tolleranza.
- 556 difettoso.
- livelli di drive non TTL.
- potenziometri R40 ed R41 difettosi.
- DS1 o DZ2 invertiti o difettosi.

Operatività

Prima di iniziare il trasferimento da cassetta a memoria, il sistema si pone in attesa di un tono basso che abbia una persistenza di almeno cinque secondi. Rilevata questa testata, che precede i dati significativi, viene passato il controllo alla routine CASTOMEM (Listing 1).

Per la scrittura viene passato il controllo alla routine MEMTOCAS, che provvede a generare la testata costituita dal tono basso persistente prima della serializzazione dei dati significativi (Listing 2).

Prima di iniziare un trasferimento di dati è necessario fissare gli indirizzi di start e stop in memoria, nelle locazioni indicate con il simbolo «$».

Il software può essere modificato per funzionare a diverse Baud rate modificando le costanti di temporizzazione. È bene ricordare che la costante base è quella riportata in *1; la costante *2 è maggiore del 50%, mentre *3 è il doppio di *1.

Conclusioni

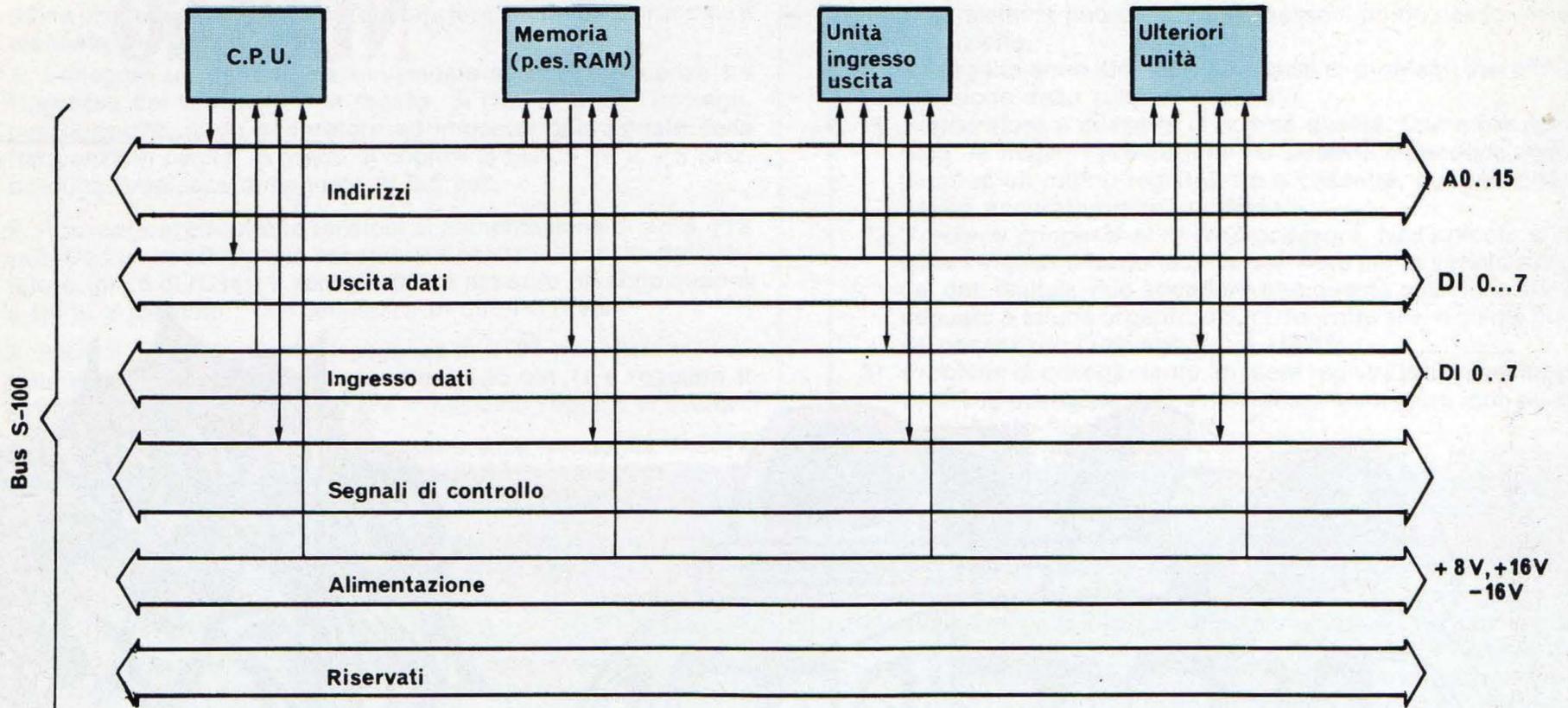
Questa interfaccia consente di realizzare un semplice e affidabile supporto per la memorizzazione di dati e programmi; altre applicazioni possono essere ottenute senza apportare modifiche allo schema base. Inoltre lo schema è indipendente dal tipo di microprocessore impiegato.

■

Il BUS S-100: Uno standard «de facto» sul mercato dei microcomputers

di **Bernd Pal**

Per nessun altro microcomputer si trovano sul mercato tante unità funzionali ad un prezzo così basso come per il cosiddetto Bus S-100. Malgrado alcune insufficienze tecniche, questo Bus è un'interessante alternativa non solo per l'hobbysta, ma anche per il progettista di professione, specialmente se sono previsti ampliamenti futuri.

Figura 1 — Organizzazione del Bus S-100.

Cronistoria

Nel gennaio 1975 la ditta americana MITS ha lanciato sul mercato un kit per microcomputer basato sul microprocessore Intel 8080, nominato Altair, al prezzo di 400 dollari. Era il primo microcomputer accessibile ad una larga cerchia di persone e si inseriva in una fetta di mercato formata da appassionati di elettronica che volevano mettere in pratica le loro conoscenze teoriche. L'Altair fu, come si dice, un successo «bomba» e fece nascere negli USA un gran numero di computers per hobbysti.

Stimolati dal successo dell'Altair, altri costruttori hanno realizzato un proprio computer basato sul Bus S-100 o più semplicemente schede di sistema ad esso compatibili. Attualmente negli USA vi sono ben 100 costruttori, dei quali almeno 5 di microcomputers completi che fanno riferimento al Bus S-100.

Figura 2 — Numerazione dei contatti sui connettori.

51	52	53	-----------	98	99	100
1	2	3	-----------	48	49	50

La gamma di produzione è estremamente ampia in quanto raggruppa sia unità standard come CPU, schede di ingresso/uscita o di memoria (fino a 64 Kbyte su una scheda!) sia schede per impieghi speciali.

Il sistema

Il sistema realizzato dalla MITS ha una struttura modulare che consente di collegare le unità funzionali (CPU, memorie, unità ingresso/uscita) tramite un bus su cui corrono indirizzi, dati e segnali di controllo. Tale sistema è stato denominato «Bus S-100», che significa «Standard 100», ed indica che si hanno a disposizione 100 collegamenti sul bus. Tali collegamenti si possono dividere in 6 gruppi (fig.1): bus indirizzi, bus dati in ingresso, bus dati in uscita, bus segnali di controllo, alimentazione e collegamenti di riserva. Le schede del si-

Errata-Corrige

1) L'architetto Fulvio Pari di Carpenedolo (BS) ritiene che l'istruzione LETX = INT (A*RND+B) comparsa nel n. 1 di Bit a pag. 26, non sia generale e che la formulazione più esatta sia:

LETX = INT (A=B+*RND+B)

2) La prima pagina dell'articolo sul microprocessore 9440 della Fairchild (pag. 55 del n. 1 di Bit) è stata composta in maniera errata.
L'inizio dell'articolo è in terza colonna quindi ricontinua in seconda colonna e poi in prima!

Tabella 1 — **Definizioni dei segnali sul Bus S-100.**

PIN	DIREZIONE *	DENOMINAZIONE
01		+8 V
02		+16 V
03	I	XRDY
04	(I)	VI 0
05	(I)	VI 1
06	(I)	VI 2
07	(I)	VI 3
08	(I)	VI 4
09	(I)	VI 5
10	(I)	VI 6
11	(I)	VI 7
12...17		liberi
18	I	STA DSB
19	I	C/C DSB
20	(U)	UNPROT
21	(U)	SS
22	I	ADD DSB
23	I	DO DSB
24	U	Φ2
25		Φ1
26	U	PHLDA
27	U	PWAIT
28	U	PINTE
29	U	A5
30	U	A4
31	U	A3
32	U	A15
33	U	A12
34	U	A9
35	U	DO 1
36	U	DO 0
37	U	A10
38	U	DO 4
39	U	DO 5
40	U	DO 6
41	I	DI 2
42	I	DI 3
43	I	DI 7
44	U	SM1
45	U	SOUT
46	U	SINP
47	U	SMEMR
48	U	SHLTA
49	U	CLOCK
50		massa
51		+8 V
52		−16 V
53	I	SSW DSB
54	U	EXT CLR
55	(U)	RTC
56...67		libero
68	U	MWRT
69	(I	PS
70	(U)	PROT
71	(U)	RUN
72	I	PRDY
73	I	PINT
74	I	PHOLD
75	I	PRESET
76	U	PSYNC
77	U	PWR
78	U	PDBIN
79	U	A0
80	U	A1
81	U	A2
82	U	A6
83	U	A7
84	U	A8
85	U	A13
86	U	A14
87	U	A11
88	U	DO 2
89	U	DO 3
90	U	DO 7
91	I	DI 4
92	I	DI 5
93	I	DI 6
94	I	DI 1
95	I	DI 0
96	U	SINTA
97	U	SWO
98	U	SSTACK
99	I	POC
100		massa

* visto dalla scheda C.P.U.: I = ingresso, U = Uscita.

Tra parentesi i segnali assenti su alcune C.P.U.

stema si collegano al bus attraverso connettori a 2x50 poli ad inserzione diretta. La numerazione dei contatti è rappresentata in fig. 2.

La tabella 1 dà una panoramica della distribuzione dei segnali sul bus.

Collegamenti bus unidirezionali

Una caratteristica importante di questo progetto è che tutti i collegamenti sul bus sono unidirezionali. Ciò significa che dati e segnali riferiti alla scheda CPU, vanno solo in un senso: o dalla CPU alle altre unità (collegamento in uscita, U) oppure da queste alla CPU (collegamento in ingresso, I).

Bus dati

L'unidirezionalità ha delle importanti conseguenze per quanto riguarda i dati del «Bus S-100».

Nella maggior parte degli altri progetti il bus dati è bidirezionale, serve perciò alla CPU sia per leggere che per scrivere dati. Il sistema S-100 comporta la suddivisione del bus dati in 2 canali da 8 collegamenti ciascuno: un bus d'ingresso dati (Di ∅...7) attraverso il quale la CPU riceve dati ed un bus d'uscita dati (DO ∅...7), attraverso il quale la CPU invia i dati alle altre unità. I vantaggi di una tale soluzione sono evidenti: da un lato si semplifica il controllo temporale dei dati (sono ridotti i tempi critici durante i quali i dati devono essere stabiliti sul bus) e dall'altro è semplificato l'accesso ai dati, di tipo DMA. Questa soluzione permette inoltre di ridurre notevolmente i segnali spuri dovuti a processi transitori sui collegamenti del bus in corrispondenza della commutazione tra dati di lettura e scrittura.

Segnali di controllo

Sul bus è definita una quantità apparentemente molto grande di segnali di controllo (44 collegamenti) che permettono il colloquio tra le singole unità e la CPU, la gestione di Interrupts su 8 livelli e l'accesso dall'esterno alla scheda del sistema evitando la CPU.

In prima approssimazione si possono riassumere in 8 gruppi.

Segnali di sincronismo

Questi quattro segnali si dividono in 2 sottogruppi (tabella 2).

Il primo serve direttamente alla sincronizzazione della CPU con le altre unità. È costituito dai segnali Φ_1 e Φ_2 del processore che vengono derivati, opportunamente disaccoppiati, dal generatore di sincronismo. (Non sempre nei progetti della CPU vengono portati sul bus entrambi i segnali di sincronismo Φ_1 e Φ_2: ciò dipende dall'impiego di generatori di clock integrati (p. es. 8224) che forniscono solo Φ_2 a livelli logici TTL. Dato che alcune schede del sistema necessitano di entrambi i segnali di sincronismo, si possono avere in questi casi

Tabella 2 — **Segnali di sincronismo.**

PIN	DENOMINAZIONE	SIGNIFICATO
25	Φ 1 phase 1 clock	Segnale di sincronismo, fase 1.
24	Φ 2 phase 2 clock	Segnale di sincronismo, fase 2.
49	CLOCK chrystal clock	Segnale di sincronismo dell'oscillatore al quarzo.
55	RTC real time clock	Segnale di clock in tempo reale.

OGGETTO: CLZ80 MICROCOMPUTER SYSTEM BASATO SULLA CPU Z80
OPERAZIONE: EUROPA 1979
PROGRAMMA: PRODUZIONE E DISTRIBUZIONE DI UN SISTEMA DI SCHEDE VERAMENTE EUROPEO ASSISTENZA TECNICA E APPLICATIVA

La SGS-ATES, per i risultati conseguiti in dieci anni di attivo impegno nello sviluppo di tecnologie MOS, è stata scelta dalla ZILOG come unico produttore europeo dello Z80, il più potente microprocessore a 8 bit esistente sul mercato.

Sulla base di questa esperienza, la SGS-ATES ha sviluppato un sistema di schede microcomputer in formato doppio Eurocard.

È un sistema completo, comprensivo di hardware, software e periferiche. Sue caratteristiche fondamentali sono la modularità e la totale espandibilità, che consentono di ampliarlo in qualsiasi momento, salvaguardando interamente gli investimenti precedenti.

Una rete capillare di distribuzione e di assistenza assicura la sostituzione immediata delle schede guaste. Non è stato trascurato il problema del training di chi non sia già esperto utilizzatore di microcalcolatori. Lo si è risolto realizzando appositamente un terminale e un software adeguati, che si integrano in un sistema didattico veramente alla portata di tutti.

Programma a misura europea

SGS-ATES Componenti Elettronici S.p.A. - Via C. Olivetti, 2 - 20041 Agrate B. - Tel.: 039-650341 - Telex: 330131 - 330141

dei malfunzionamenti). Il secondo gruppo di segnali ha compiti molto più generali; è costituito dal segnale primario dell'oscillatore al quarzo (col processore 8080A principalmente 2 MHz), che può servire a unità periferiche per generare segnali di sincronismo interni, e da un segnale di «Real Time Clock» (RTC). Quest'ultimo viene generato normalmente da una speciale scheda del sistema.

Ingressi diretti al processore

In questo caso si tratta dei due segnali di Ready, che indicano la disponibilità delle schede del sistema allo scambio dei dati con la CPU, e delle richieste d'interruzione Interrupt-Request e Hold-Request (tabella 3). I due segnali di Ready, PRDY del sistema stesso e XRDY delle unità esterne, vengono collegati in AND sulla scheda CPU e portati al pin 23 del processore 8080, mentre le richieste d'interruzione (attraverso invertitori) vengono collegate direttamente con i pins 13 (HOLD) e 14 (INT).

Segnali di comando e controllo del processore

Questi segnali sono generati direttamente dal processore e sono disponibili ai pins 16 (INTE), 17 (DBIN), 18 ($\overline{WR}$), 19 (SYNC), 21 (HLDA) e 24 (WAIT). Vengono portati dalla CPU al bus attraverso drivers. Con essi la CPU sincronizza il traffico dei dati con le schede del sistema (tabella 4).

Segnali di stato

Per lo scambio dati il processore dà, all'inizio di ogni ciclo macchina, una parola di stato sul bus dei dati. Questa parola di stato viene decodificata sulla scheda CPU e memorizzata in un registro a 8 bit le cui uscite sono portate al bus attraverso drivers. I segnali di stato sono indicati in tabella 5 preceduti dal prefisso S. Tali segnali sono stabili durante tutto il ciclo macchina, e quindi non devono venire decodificati continuamente dalle schede del sistema. Come conseguenza si ha lo scarico del bus dei dati d'uscita ed una semplificazione nel progetto delle restanti schede del sistema, che non devono decodificare tutte le volte nuovamente lo stato CPU.

Segnali di controllo della memoria

Questo gruppo è costituito da 4 segnali (tabella 6) di cui solo uno (MWRT, Memory Write Enable) deve essere necessariamente portato alla scheda CPU. Gli altri tre possono venir usati da una console di comando o da una speciale scheda del sistema.

Questi ultimi segnali offrono la possibilità di proteggere la memoria RAM da sovrascritture. Per permettere ciò, la scheda RAM deve essere opportunamente predisposta. La protezione del blocco di memoria prescelto (sono usuali passi di 256 Bytes... profondità 4 K Bytes) viene effettuata mettendo l'indirizzo d'inizio del blocco da proteggere sul bus d'indirizzo e alzando con-

Tabella 3 — Ingressi diretti al processore.

PIN	LOGICA	DENOMINAZIONE		SIGNIFICATO
72	pos.	PRDY	processor ready	Segnale di disponibilità del sistema.
3	pos.	XRDY	external ready	Segnale di disponibilità delle unità esterne.
74	neg.	$\overline{PHOLD}$	processor hold request	Richiesta di hold.
73	neg.	$\overline{PINT}$	processor interrupt request	Richiesta d'interruzione.

Tabella 4 — Segnali di comando e di controllo del processore.

PIN	LOGICA	DENOMINAZIONE		SIGNIFICATO
28	pos.	PINTE	processor interrupt enable	Segnala che è stato abilitato il flip-flop «Interrupt-enable».
78	pos.	PDBIN	processor data bus in	Indica che il bus dei dati può ricevere i dati dalla unità indirizzata.
77	neg.	$\overline{PWR}$	processor writes	Indica che i dati emessi dal processore sono stabili.
76	pos.	PSYNC	processor synchronize	Dà l'inizio di ogni ciclo macchina.
26	pos.	PHLDA	processor hold acknowledge	Segnala che la C.P.U. è nello stato Hold.
27	pos.	PWAIT	processor wait acknowledge	Segnala un ciclo di attesa del processore.

Tabella 5 — Segnali di stato.

PIN	LOGICA	DENOMINAZIONE		SIGNIFICATO
96	pos.	SINTA	status bit 0: interrupt acknowledge	Segnala che è stata ricevuta una richiesta di interruzione.
97	neg.	$\overline{SWO}$	status bit 1: write operation	Segnala che viene eseguita una operazione di scrittura.
98	pos.	SSTACK	status bit 2: stack addressed	Segnala che l'indirizzo di stack si trova sul bus indirizzi.
48	pos.	SHLTA	status bit 3: halt acknowledge	Segnala che è stato ricevuto un comando HLT.
45	pos.	SOUT	status bit 4: output device addressed	Segnala che il bus indirizzi porta l'indirizzo di una unità di uscita.
44	pos.	SM1	status bit 5: machine cycle 1	Segnala l'inizio di un ciclo di comando (ciclo macchina 1).
46	pos.	SINP	status bit 6: input device addressed	Segnala che il bus indirizzi porta l'indirizzo di una unità d'ingresso.
47	pos.	SMEMR	status bit 7: memory read cycle	Segnala che viene eseguito un ciclo di lettura della memoria.

temporaneamente il segnale PROT (Protect). In questo modo si setta un flip-flop sulla scheda di memoria che blocca attraverso una porta l'impulso di scrittura MWRT.

La protezione viene disinserita per mezzo del segnale UNPROT.

L'indirizzamento di una memoria protetta viene segnalato attraverso il collegamento PS (generalmente verso la console). Il sistema viene inizializzato in maniera tale che all'accensione venga annullata ogni protezione di memoria.

Segnali di comando per l'accesso al bus dall'esterno.

I due bus dati, il bus indirizzo e i segnali di controllo (tabelle 4 e 5) fanno capo alla CPU tramite drivers tri-state.

Questi drivers sono messi nello stato di alta impedenza dai 5 segnali di controllo di questo gruppo (tabella 7), permettendo così l'accesso al bus dall'esterno. Il nome dei 5 segnali di comando indica i drivers su cui il segnale interviene.

Il segnale SSW DSB mette in tri-state il bus dati in ingresso permettendo l'introduzione dei dati dalla scheda console tramite interruttori (Altair: sense switch).

Segnali di Interrupt

Questi segnali generalmente non vanno alla scheda CPU. Con essi si controllano diversi livelli d'interruzione (Vectored Interrupt), che devono venir decodificati da un'unità separata (nella maggior parte dei casi è sistemata su una apposita scheda del sistema) che genera in corrispondenza della priorità una richiesta d'interruzione sul collegamento PINT.

Ulteriori segnali di controllo

Possono essere riuniti in 2 gruppi (tabella 8). Il primo gruppo è composto dai segnali d'inizializzazione della macchina, inizializzazione all'accensione (POC), inizializzazione della CPU durante il funzionamento normale (PRESET) e inizializzazione delle unità d'ingresso ed uscita (EXT CLR).

Livelli logici e caricabilità del bus

Tutti i collegamenti sul bus relativi a segnali hanno livelli TTL. Normalmente si usa la logica positiva (1 = H, O = L). Su alcuni collegamenti di controllo anche la logica negativa (1 = L, 0 = H). Nelle tabelle 2...8 viene indicata la logica impiegata. La maggior parte dei collegamenti possono portare, attraverso i drivers sulle schede, fino a 10 ingressi TTL Standard o 30 ingressi Low-Power-Schottky-TTL. Gli ingressi delle schede corrispondono al massimo ad un carico TTL-Standard. Il «fan-out» è perciò sufficiente per realizzare, a seconda del progetto, un sistema completo di non oltre 25 schede.

Alimentazione

Il sistema prevede dei regolatori di tensione sulle singole schede. Si eliminano così accoppiamenti non desiderati attraverso i collegamenti di alimentazione. Inoltre risulta sufficiente un'alimentazione centrale semplice non stabilizzata.

Tabella 6 — Segnali di controllo della memoria.

PIN	LOGICA	DENOMINAZIONE		SIGNIFICATO
68	pos.	MWRT	memory write enable	Impulso di scrittura in memoria.
70	pos.	PROT	protect memory area	Protezione di una zona della memoria da sovrascritture.
20	pos.	UNPROT	unprotect memory area	Sospensione protezione dalla scrittura.
69	neg.	PS	protect status	Segnala che la zona di memoria indirizzata è protetta.

Tabella 7 — Accesso dall'esterno.

PIN	LOGICA	DENOMINAZIONE		SIGNIFICATO
18	neg.	STA DSB	status buffer disable	Disabilita i drivers di stato sulla scheda C.P.U.
19	neg.	C/C DSB	command/control buffer disable	Disabilita i drivers di comando e controllo sulla scheda C.P.U.
22	neg.	ADD DSB	address buffer disable	Disabilita i drivers di indirizzo sulla scheda C.P.U.
23	neg.	DO DSB	data out buffer disable	Disabilita i drivers dei dati di uscita sulla scheda C.P.U.
53	neg.	SSW DSB	sense switch buffer disable	Disabilita i drivers dei dati di ingresso sulla scheda C.P.U.

Tabella 8 — Ulteriori segnali di controllo.

PIN	LOGICA	DENOMINAZIONE		SIGNIFICATO
99	neg.	POC	power on clear	Inizializza il sistema all'inserimento dell'alimentazione.
75	neg.	PRESET	processor reset	Segnale di reset per il processore e la logica di C.P.U.
54	neg.	EXT CLR	external clear	Inizializza le unità Ingresso/Uscita.
21	pos.	SS	single step mode	Indica che il programma è eseguito passo passo.
71	pos.	RUN	run mode	Segnala che il programma è in esecuzione.

Sul bus sono presenti le seguenti tensioni:
+ 8 V sui collegamenti 1 e 51
+16 V sul collegamento 2
−16 V sul collegamento 52
massa sui collegamenti 50 e 100

L'alimentazione 8 V viene abbassata alla tensione standard + 5 V sulla scheda: serve principalmente ad alimentare la logica TTL ed è necessaria per ogni scheda del sistema. Le altre due tensioni servono soprattutto ad alimentare i componenti MOS. Anch'essi vengono abbassati al valore richiesto sulla scheda.

Un sistema, completamente utilizzato, necessita di ca. + 8 V/10...20A, +16 V/2...5 A e −16 V/1 A.

Collegamenti di riserva

Alimentazioni, indirizzi, dati e segnali di controllo occupano 82 dei 100 collegamenti sul bus. I restanti 18 collegamenti restano liberi per gli impieghi speciali e per le modifiche del sistema.

Vi sono però già proposte di standardizzazione per microcomputers a 16 bit (sulla base del TMS 9900 della Texas Instruments) e per nuovi tipi di memoria come CCD e memorie a bolle magnetiche, che necessitano di ulteriori segnali di controllo.

Discussione

Non esiste (ancora) il bus ideale, e di conseguenza vie è una serie completa di progetti alternativi al Bus S-100 che tendono ad ovviare alle sue carenze. Alcuni dei punti deboli del sistema sono riportati qui di seguito:

— Dalla panoramica della tabella 1 si nota che i collegamenti d'indirizzo e dei dati sono vicinissimi. Ciò provoca accoppiamenti spurii che tendono ad aumentare il rumore di fondo che non può sempre essere trascurato. La stessa cosa vale per i due segnali di sincronismo Φ_1 e Φ_2. Si può ovviare a ciò introducendo uno schermo di massa tra i due blocchi di segnali come viene già fatto da alcuni costruttori sulla «motherboard» del bus.
— Il sistema della regolazione di tensione distribuita ha lo svantaggio che la sollecitazione termica tra le schede spesso aumenta considerevolmente, e ciò può causare il cedimento dei componenti. Nell'assemblaggio del sistema occorre quindi curare in modo particolare, più che in altri progetti, la dispersione del calore. Il surriscaldamento diventa tanto più grande quanto più sono piccoli i dissipatori a causa della vicinanza delle schede e quindi i piccoli sistemi richiedono un ventilatore per un funzionamento sicuro.
— Il Bus S-100, in quanto porta sul bus la parola stato e i segnali di controllo del processore 8080, è altamente specializzato. Se si vogliono sfruttare i vantaggi economici e tecnici, dovuti al gran numero di ditte che costruiscono attorno al Bus S-100, per utilizzare altri microprocessori (p. es. 6800, 6502) o ulteriori sviluppi dell'8080 (p. es. Z-80), sono necessari complessi adattamenti per i segnali di controllo. Bisogna considerare però che per i suddetti processori esistono già progetti adatti al Bus S-100.

Prospettive

Il Bus S-100 ha preso piede e in maniera tale che alcuni costruttori offrono due versioni dei loro progetti, uno per il Bus Intel ed uno per il Bus S-100. Malgrado tutti gli svantaggi del progetto, è opportuno tener conto del Bus S-100 in Europa per diversi motivi:
— La grande quantità di prodotti offerti per questo sistema, come già detto.
— Il fatto che la forte concorrenza soprattutto di piccoli e medi costruttori abbassa naturalmente i prezzi e ciò provoca la continua apparizione di prodotti ad alto livello e tecnicamente avanzati sul mercato.
— Last not least, non si deve dimenticare che il microprocessore 8080 è stato costruito su uno standard industriale che a detta degli esperti dovrebbe durare possibilmente due decenni.

■

A proposito dell'S-100

Prima di tutto non è corretto affermare che l'S-100 è un «bus standard» perchè è stato progettato attorno al microprocessore 8080 e solo successivamente rimaneggiato per altri micro, per lo più sempre del filone 8080 (8085, Z 80). È comunque poco conveniente interfacciare a questo bus altre validissime macchine, quali 6800, 2650, 1802, SC/MP. La paternità 8080 è rispecchiata nei segnali di controllo; le linee PSYNC, PINTE, PWAIT, SSTACK, sono tipiche di questo microprocessore e non è possibile ricavare segnali con lo stesso significato dagli altri. Per contro il Bus S-100 non prevede (almeno nella proposta IEEE 696.1/D2) linee per altri segnali ormai di uso comune, quali l'interruzione non mascherabile (NMI), l'indicazione di ciclo di rinfresco per le memorie dinamiche, (RFR) l'arbitraggio di DMA multipli. La struttura del bus dati non bidirezionale impone di usare sempre dei separatori (buffer), anche in sistemi a configurazione minima, mentre nei componenti di uso corrente la bidirezionalità delle linee dati è ormai un fatto acquisito. I tentativi di estendere l'S-100 a processori con 16 bit di dati e più di 16 linee di indirizzo conducono a definizioni accuratamente incompatibili con lo «standard» attuale (linee dati bidirezionali o multiplexate con gli indirizzi).

È da notare che già adesso alcune linee sono usate per funzioni diverse dai vari costruttori, e che la compatibilità non è poi così totale come si potrebbe pensare. A questo punto lo «standard S-100» diventa niente più che una scelta delle dimensioni di scheda e del tipo di connettore.

Con tutto questo all'S-100 va riconosciuto un innegabile merito: quello di *esistere* e di essere un punto di riferimento per costruttori ed utenti, creando per i secondi una possibilità di scelta che i «grandi» (Texas, Intel, Motorola e gli altri tanto per fare degli esempi), non hanno voluto dare.

Il fenomeno S-100 resta quindi comunque una pietra miliare per lo sviluppo dei microcalcolatori «indipendenti», e sarà mantenuto sulla breccia ancora per molto tempo.

È essenziale non tanto accettare il contenuto strettamente tecnico dello «standard S-100» e subire una scelta che un altro mercato ha già fatto nella notte dei tempi (la scala dei tempi è rapportata alla velocità dei micro!), quanto cogliere la dimostrazione di cosa è possibile fare con un minimo di buona volontà da parte dei costruttori, a tutto vantaggio loro e degli utenti.

Ing. D. Del Corso
Istituto di Elettronica
Politecnico di Torino

Bibliografia

Zaks, R: Microprocessors. Sybex Inc. Berkeley, Cal. Edizione Italiana Jackson Italiana s.r.l.

Lavorare in Basic

a cura della redazione

Cassetta magnetica e floppy-disk sono le memorie di massa tipiche di un personal computer. Con quali istruzioni BASIC, con quali tecniche, per quali applicazioni vengono utilizzate?

Il linguaggio BASIC usato su molti personal computers in origine era stato sviluppato al Dartmouth College per grandi sistemi centralizzati time-shared; l'obbiettivo che si voleva raggiungere nel definire il BASIC era rendere la programmazione facile anche per gli inesperti. Ed è proprio per le sue caratteristiche di semplicità che il BASIC è così popolare.

Per una descrizione delle istruzioni tipiche del BASIC si rimanda all'articolo «Introduzione al Basic», apparso sul numero 1 di BIT.

In figura 1 è riportato come esempio un programma che calcola il numero di giorni che intercorrono tra la propria data di nascita e la data corrente.

Le istruzioni BASIC più complesse sono quelle che permettono l'Input-Output, ed è proprio in queste istruzioni che le diverse versioni del linguaggio BASIC variano enormemente. Classifichiamo le istruzioni di I/O in due categorie:

1) I/O su terminali
2) I/O su memoria di massa.

Terminali

Le istruzioni di I/O su terminale sono usate per comunicare tra l'utente ed il calcolatore; terminali tipici sono la tastiera e la stampante.

Figura 1 — Programma del calendario.

Il programma del calendario, sezione (A) del listing, considera gli anni bisestili e i differenti numeri di giorni per ogni mese.

La data di partenza nel programma è il 1° gennaio 1900, per cui l'anno è specificato dando le ultime due cifre.

La soubroutine che converte gli anni-mesi-giorni in giorni, sezione (B) del listing, utilizza fondamentalmente la funzione INT, con cui si tiene conto della diversa lunghezza dei mesi (28, 29, 30 o 31 giorni) e degli anni bisestili (la linea 1200 contiene la costante 146), che è il numero di giorni contenuti in 4 anni, di cui uno bisestile.

(A) Programma principale

```
100  REM     ==============================
110  REM
120  REM     DA QUANTI GIORNI SONO NATO?
130  REM
140  REM     ==============================
150  REM
160  PRINT   "QUANDO SEI NATO?"
170  PRINT   "INTRODUCI IL MESE:";
180  INPUT   J
190  PRINT   "QUALE GIORNO?";
200  INPUT   K
210  PRINT   «QUALE ANNO 19»;
220  INPUT   I
230  LET     I = I + 1900
240  REM
250  REM     --- CHIAMA IL CALENDARIO DEI GIORNI -
260  REM
300  GOSUB   1000
305  LET     J9 = J0
310  REM
320  REM     --- CALCOLA LA DIFFERENZA ---
325  REM
330  PRINT   "CHE GIORNO È OGGI?"
340  PRINT   "INTRODUCI IL MESE:";
350  INPUT   J
360  PRINT   "CHE GIORNO?";
370  INPUT   K
380  PRINT   "INTRODUCI L'ANNO 19";
390  INPUT   I
400  LET     I = I + 1900
410  REM
420  REM     --- CALCOLA I GIORNI ---
430  REM
435  GOSUB   1000
440  LET     L = J0 – J9
450  PRINT   USING 500 : L
460  END
500  IMAGE   "HAI VISSUTO", 7D, "GIORNI — BUONA
             CONTINUAZIONE!"
```

(B) Subroutine per il calcolo dei giorni trascorsi

```
1000  REM     ==============================
1010  REM
1020  REM     NOTA LA DATA IN MESE-GIORNO-ANNO,
              CALCOLA QUELLA IN GIORNI
1030  REM     PARTENDO DAL PRIMO GENNAIO 1900
1040  REM
1050  REM     ==============================
1060  REM
1070  REM     J = MESE
1080  REM     K = GIORNO DEL MESE
1090  REM     I = Anno (4 CIFRE)
1100  REM
1160  LET     J1 = 0
1170  IF      J > 2 THEN 1190
1180  LET     J1 = –1
1190  LET     J2 = INT (I + 4800 + J1)
1200  LET     J2 = INT (1461 * J2)
1210  LET     J2 = INT (J2 / 4)
1220  LET     J3 = INT (K –32075 + J2)
1230  LET     J4 = INT (J–2–J1 *12)
1240  LET     J4 = INT (367 *J4)
1250  LET     J4 = INT (J4 / 12)
1260  LET     J5 = INT ((I +4900 + J1) / 100)
1270  LET     J5 = INT (3 =J5)
1280  LET     J5 = INT (J5 / 4)
1290  LET     J0 = J3 + J4 – J5 – 2415021
1300  REM
1310  REM     ---PRIMO GENNAIO 1900 È IL GIORNO
              2415021 ---
1370  RETURN
```

(C) Esempio di esecuzione del programma presentato

```
QUANDO SEI NATO?
INTRODUCI IL MESE: 12
QUALE GIORNO? 2
QUALE ANNO 1941
CHE GIORNO È OGGI?
INTRODUCI IL MESE: 6
CHE GIORNO? 29
INTRODUCI L'ANNO 1977
HAI VISSUTO 12993 GIORNI — BUONA CONTINUAZIONE!
```

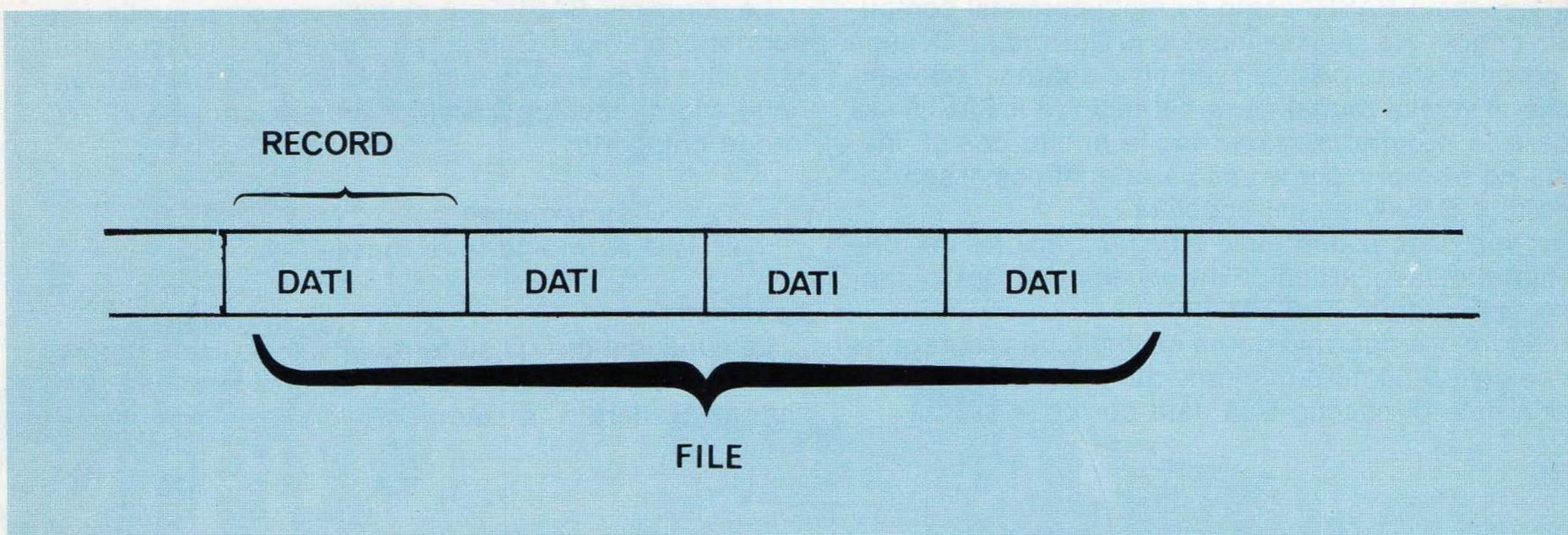



Figura 2 — Rappresentazione dei dati su un nastro magnetico.

Le operazioni di I/O su terminale portano con sè il problema del formato, cioè di quale è la notazione da usare per inserire o far stampare un dato.

Consideriamo le seguenti istruzioni:

```
105 LET X = 1.0
115 PRINT X
```

La maggior parte dei personal computers esegue la sequenza indicata sopra stampando

0.100000 E 1

o una notazione simile, equivalente a 1.0.

Questa modalità di stampa può essere scomoda soprattutto quando il numero da scrivere ha un significato particolare, ad esempio richiede due cifre a destra del punto decimale.

Il formato di stampa può essere specificato in BASIC dando l'istruzione.

PRINT USING [identificatore di linea] : [variabile] ([1]) (o espressioni equivalenti) associata, tramite l'identificatore di linea, all'istruzione:

IMAGE [format]

La sequenza scritta prima può venire riscritta come segue:

```
105 LET X = 1.0
115 PRINT USING 120 : X
120 IMAGE 3D.2D
```

Cosi come l'istruzione PRINT, anche IMAGE può differire nel formato da una versione di BASIC ad un'altra.

Nell'esempio precedente, otteniamo la stampa di 1.00, poichè 3D.2D significa mettere due cifre a destra e tre a sinistra del punto decimale. Si noti che l'istruzione 115 usa la 120 come istruzione di formato.

Per concludere questa breve presentazione dei formati, possiamo aggiungere che è possibile indirizzare l'output su un particolare terminale, usando le istruzioni seguenti:

```
115 PRINT @33 : USING 120 : Z
120 IMAGE 10X, 3D. 2D
```

dove «@33:» individua una stampante a cui è stato associato il numero 33 prima di lanciare il programma in esame. Il formato 10X fa apparire 10 spazi prima di Z.

Memoria di massa

Le istruzioni di I/O su memoria di massa permettono l'accesso ad un nastro magnetico o ad un disco. Nei dischi e nei nastri magnetici i dati sono organizzati in files. Un *file*, a sua volta, contiene uno o più records, essendo il *record* un insieme di caratteri alfanumerici che costituiscono l'informazione minima a cui si può accedere (vedi figura 2).

In BASIC, dobbiamo leggere o scrivere un record alla volta.

L'input-output su nastri e dischi è profondamente diverso in BASIC dall'input-output su terminale; il motivo è molto semplice: nastri magnetici e dischi operano in modo differente, e quindi devono venire trattati in modo differente.

Un nastro è sequenziale, cioè non si può accedere ad un record all'interno del file senza aver prima letto tutti i records che lo precedono, a partire dall'inizio del file stesso.

Possiamo dire che in un file su nastro un record viene accodato a quello scritto immediatamente prima; per modificare il secondo record, ad esempio, dobbiamo scrivere un programma che legga prima il primo record e poi il secondo.

Un'altra restrizione sull'uso dei nastri è che per cambiare un record su nastro, dobbiamo riscrivere l'intero file.

Riscrivere un record, infatti, richiederebbe una coincidenza perfetta tra la posizione del vecchio e quella del nuovo record, ma, dal momento che una unità nastro non è accurata a tal punto da riscrivere esattamente un record nella stessa posizione di quello preesistente, si rende necessario ricopiare l'intero file su un nuovo nastro.

In un disco (vedi figura 3) i records sono memorizzati in settori. Un *settore* di disco è come un nastro in miniatura, perchè si comporta esattamente come un nastro. Perciò, quando un record, ad esempio A B C, è scritto su un disco in un settore, è possibile aggiornarlo accedendo in *modo diretto* ad esso.

Sfortunatamente, un settore di disco può essere troppo grande o troppo piccolo rispetto alle dimensioni del nostro record; in tal caso, o sprechiamo spazio, o dobbiamo spezzare i records, una parte in un settore ed un'altra parte nel settore adiacente.

Per accedere ai files su memoria di massa devono venire espletate particolari funzioni, dopo di che è possibile la lettura e/o scrittura, e cioè:

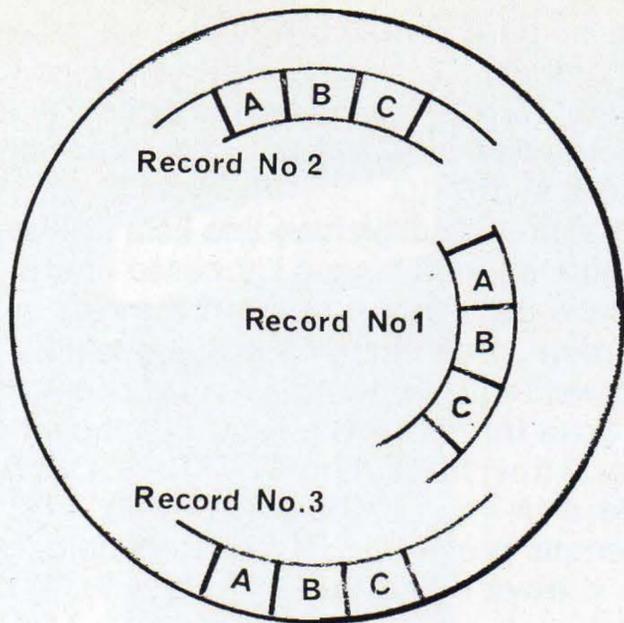



Figura 3 – Struttura di un file sequenziale su disco.

1) OPEN o FIND: questa operazione è eseguita prima di qualunque altra operazione di I/O, in modo da stabilire se il file, a cui spesso è associato un nome logico, è presente nel disco o nastro:

```
OPEN # 1, "DATA"
```

In questo esempio il file di nome DATA presente sulla unità numero 1 è preparato alla lettura/scrittura.

Nel caso di nastro magnetico è usata solitamente l'istruzione:

```
FIND I
```

In questo caso si accede al file in posizione I-esima dall'*inizio del nastro.*

2) RESTORE, %, o FIND (solo su disco): questa operazione è usata su disco quando si vuole accedervi direttamente (cioè in modo non sequenziale). Il risultato di un RESTORE è posizionare il meccanismo di lettura/scrittura del disco direttamente sopra il settore o il record che deve essere letto o scritto. I settori sono numerati da zero ad un certo massimo.

Per esempio:

```
RESTORE # 1, 58
```

prepara alla lettura o scrittura il settore numero 58 del file 1.

3) CLOSE o END: questo comando chiude un file aperto (tramite OPEN o equivalente). Spesso, il comando CLOSE o END scrive un carattere speciale di *end-of-file* (EOF) su nastro o disco, che sta ad indicare che si è arrivati alla fine del file.

Per dettagli maggiori sulle operazioni descritte sopra è necessario ricorrere ai manuali della specifica versione di BASIC che si ha a disposizione.

Un esempio di utilizzo di file su nastro è il seguente:

```
2112 FIND 2
2120 ON EOF (0) THEN 2350
2130 INPUT @33 : I, A$, B$, C$
2140 GOTO 2120
```

Questo pezzo di codice mostra come cercare la fine del file su nastro.

La funzione EOF (0) dà come risultato il valore TRUE se si è raggiunto l'end-of-file.

Si noti l'istruzione INPUT: «@33:» individua la periferica da cui avviene l'input di dati, mentre la sequenza di simboli I, A$, B$, C$ indicano le variabili a cui i dati letti sono assegnati. In particolare è specificato che un record è formato da un intero e da tre stringhe alfanumeriche.

Un metodo simile di ricerca su disco ma con accesso diretto è il seguente:

```
2105 LET I1 = 0
2110 OPEN # 2, "DFILE"
2120 ON EOF (2) THEN 2350
2130 RESTORE # 2, I1
2135 READ # 2, I, A$, B$, C$
2140 LET I1 = I1 + 1
2145 GOTO 2120
```

Questo esempio mostra come simulare un file sequenziale su disco. La variabile I1 punta sempre il settore del disco che contiene il record numero I1.

Il significato dell'istruzione READ è il seguente: comanda la lettura dei dati dal file 2 e li assegna alle variabili I, A$, B$, C$, in cui le ultime tre indicano, al solito, stringhe alfanumeriche.

Metodi di ricerca

La ricerca di una informazione in un archivio (ricerca di un record in un file) è tipica in un sistema dotato di memoria di massa. Questo problema può essere risolto in vari modi; qui di seguito ne presentiamo alcuni tra i più comuni.

Ricerca sequenziale per chiave

È uno dei metodi di ricerca più semplici e funziona piuttosto bene con i nastri magnetici.

L'algoritmo utilizzato è il seguente: *si inizia dal primo record e si prosegue con i successivi sino a quando non si è trovato quello voluto.* Per fare questo si utilizza una *chiave* di ricerca (che ad esempio può essere la prima informazione contenuta nel record); quando si sono trovate due chiavi uguali, si è raggiunto il record voluto.

Qui di seguito è riportato un pezzo di programma che illustra la strategia di *ricerca lineare:* si suppone che l'insieme di dati entro cui cercare sia rappresentato dal vettore L (in memoria), e che la chiave di ricerca consista nella variabile K (sempre in memoria).

```
 10 DIM L (100)
 .. .........
 80 INPUT K
100 FOR I = 1 TO 100
110 IF L(I) = K THEN 300
120 NEXT I
130 REM —MANCA L'ELEMENTO CERCATO—
140 PRINT "L'ELEMENTO CERCATO NON È
    PRESENTE"
150 LET I = 0
160 RETURN
300 REM ——TROVATO AL POSTO I————
310 PRINT "TROVATO AL POSTO" I
320 RETURN
```

La ricerca lineare continua a confrontare gli elementi della lista L fino a che non nè è stato trovato uno uguale a K oppure la lista L è stata esaurita.

La ricerca su nastro avviene in modo simile a quella in un vettore. Il segmento di programma seguente è la trascrizione per nastro di quello precedente; se ne notino le differenze.

```
 10 FIND 1
 15 REM —— APRE IL FILE 1 SU NASTRO ——
 .. ............
 80 INPUT K
100 FOR I = 1 TO 100
105 INPUT @33 : L
110 IF L = K THEN 300
```

```
120 NEXT I
130 REM ---MANCA L'ELEMENTO CERCATO------
140 PRINT "L'ELEMENTO CERCATO NON È PRESENTE"
150 LET I = 0
160 RETURN
300 REM --- TROVATO AL POSTO I ---
310 PRINT "TROVATO AL POSTO"; I
320 RETURN
```

Ricerca sequenziale

Se la posizione del record nel file è nota, ad esempio record 68, si ha:

```
100 FIND 5
110 FOR I = 1 TO (I1-1)
120 INPUT @33: H, M, A$, B$, C$
130 NEXT I
```

Questo pezzo di codice legge (68-1) records, posto che all'indice I1 sia stato assegnato il valore 68.

Il comando FIND 5 apre il file 5 su nastro e oltrepassa i files 1, 2, 3, 4. Una volta che il file è stato aperto, vengono letti 67 records uno alla volta. Ciascuno di questi records è scartato perchè non è quello cercato. In questo modo ci si posiziona sul record voluto, che può ora essere letto e stampato.

```
140 INPUT @33 : H, M, A$, B$, C$
150 CLOSE 5
160 PRINT
170 PRINT H, M, A$, B$, C$.
```

In queste ultime istruzioni, è stata utilizzata l'istruzione CLOSE per chiudere il file. Questa operazione è stata fatta *prima* della stampa e comporta il rilascio dell'eventuale buffer utilizzato durante le operazioni di lettura dal nastro.

Un *buffer* è una regione della memoria principale usata per memorizzare temporaneamente i records mentre si attende che vengano o scritti sulla memoria di massa (uscita) o assegnati alle variabili indicate dal programma (ingresso). I buffers hanno la funzione di compensare le differenti velocità di elaborazione tra memoria e disco o nastro.

Se scegliamo di memorizzare il nostro archivio su disco anzichè su nastro, potremo scegliere tra i metodi di ricerca con accesso diretto e quelli con accesso sequenziale.

L'accesso sequenziale su disco è identico in BASIC all'accesso sequenziale su nastro. L'economia di spazio è uno dei motivi per cui si usa l'eccesso sequenziale anche con files su disco.

Ricerca binaria

Il disco permette l'accesso diretto, cioè si può accedere direttamente a qualsiasi record di un file senza dover leggere il file dall'inizio.

Si supponga di avere, ad esempio, la seguente lista *ordinata* su disco:

```
SETTORE 0 : 18 ←———————————— START
SETTORE 1 : 25
SETTORE 2 : 32
SETTORE 3 : 33
SETTORE 4 : 35
SETTORE 5 : 36
SETTORE 6 : 42
SETTORE 7 : 49 ←———————————— STOP
```

Ciascun numero è memorizzato in un record fisico chiamato *settore*. (La maggior parte dei settori fisici contengono 128 o 256 caratteri). I settori di un disco sono numerati da 0 in su esattamente come le posizioni di un vettore in BASIC.

Quando si ha a disposizione una lista ordinata come la precedente ed è ammesso l'accesso diretto è possibile utilizzare un metodo di ricerca molto veloce, la *ricerca binaria*, il cui schema è il seguente.

Il primo elemento confrontato con la chiave K è quello che si trova in mezzo alla lista. Supponiamo che la nostra lista si trovi tra i settori START e STOP: l'elemento centrale è INT ((START + STOP) /2).

Nell'esempio precedente, l'elemento da confrontare per primo si trova nel settore INT ((0 + 7) /2), cioè nel settore 3.

Supponiamo di voler cercare K = 35. Incominciamo i confronti dal settore 3; poichè 33 è diverso da 35, dobbiamo proseguire nella ricerca. Sappiamo però che ogni record che si trova nei settori che precedono il 3 è minore di 33, quindi scartiamo tutta la prima metà della lista, visto che 35, se c'è, si trova nella seconda metà.

Poniamo quindi START = 4 e ripetiamo la ricerca calcolando un nuovo elemento centrale.

```
SETTORE 4 : 35 ←———————————— START
SETTORE 5 : 36 ←———————————— METÀ
SETTORE 6 : 42
SETTORE 7 : 49 ←———————————— STOP
```

Ancora l'elemento centrale è diverso da K; questa volta, però, 36 è più grande di K: dobbiamo quindi scartare la seconda metà della lista e proseguire nella prima. dopo aver scartato tutti i settori successivi al 5, ci rimane un solo record nella lista, così abbiamo bisogno soltanto di un confronto per trovare K = 35.

Settore 4 : 35 ← START ← METÀ ← STOP

La ricerca binaria appena descritta ha richiesto 3 confronti per testare 8 records. Questo è un miglioramento notevole rispetto alla ricerca lineare, ma ovviamente ha i suoi inconvenienti: *il file deve essere ordinato e deve essere memorizzato su un device ad accesso diretto.*

Ricerca diretta

Per poter utilizzare l'accesso diretto, a ciascun record è associato un numero: questo numero riferito al record è utilizzato in modo simile agli indici di un vettore.

Supponiamo di avere un disco con 350 settori, numerati da 0 a 349. Allochiamo i settori come segue: il record I1 viene posto nel settore I1-1.

La maggior parte delle versioni di BASIC richiede che venga specificato il numero di settore (cioè il numero associato al record) prima di una lettura o scrittura. Uno dei metodi usati si avvale del comando RESTORE, come si vede nel segmento di programma seguente, in cui il comando RESTORE appare in un formato diverso da quello esposto precedentemente:

```
90 OPEN # 35, 1
100 INPUT "ORA", H, "MINUTI", M
... .........
410 RESTORE # (I1-1)
420 INPUT # 35 : H, M, A$, B$, C$
```

```
430 CLOSE # 35, 1
440 PRINT H, M, A$, B$, C$
```

Questo segmento di programma permette l'accesso diretto al record (I1-1) memorizzato sul device 35. Un'altra versione dello stesso programma può essere il seguente:

```
 90 OPEN # 35, 1
100 INPUT "ORA": H, "MINUTI", M
... ..........
410 REM ——— RICERCA IL RECORD I1 ———
420 READ # 35; (I1-1) : H, M, A$, B$, C$
430 CLOSE # 35, 1
440 PRINT H, M, A$, B$, C$
```

Si noti che in quest'ultimo caso è stata utilizzata l'istruzione READ al posto di INPUT e di RESTORE, e che il formato di READ è diverso da quello esposto in precedenza.

Concludiamo il discorso sul metodo di ricerca diretto dicendo che le diverse versioni di BASIC possono usare forme e metodi diverse, ma i concetti qui esposti sono comunque validi nella loro generalità.

Ricerca HASH

Vogliamo ora illustrare come si possono reperire delle informazioni alfanumeriche in modo diretto.

A tal fine occorre munire di indice i dati alfabetici in un modo simile a quello visto nei paragrafi precedenti per i dati numerici.

Supponiamo di voler costruire un file i cui records sono formati da un nome ed un numero: ad esempio un nome ed il corrispondente numero telefonico. Svilupperemo, per far questo, il programma BUILD (vedi figura 4) che costruisce il file. L'input è un nome ed un numero: il nome è usato come indice per allocare il record. Una volta che è stata determinata la posizione in cui deve essere memorizzato il record, il nome ed il numero telefonico verranno memorizzati e si passerà al record successivo.

Svilupperemo anche un programma di ricerca che chiameremo LOOK-UP (vedi figura 5), il quale accetta come input un nome, e restituisce il numero di telefono associato, stampando infine sia il nome che il numero.

Ma come è possibile utilizzare le informazioni alfabetiche (un nome, in questo esempio) per accedere direttamente ad un record di un file?

Mostreremo come ciò sia possibile, usando vettori, anzichè un file su disco. Questo modo di procedere rende la descrizione più chiara, ma si ricordi che il metodo si applica a qualsiasi dispositivo diretto: infatti si devono cambiare solo le istruzioni BASIC per accedere al file.

Il primo passo per definire un metodo di accesso a files alfabetici è trasformare i nomi in numeri. Questa funzione di trasformazione, detta *funzione HASH,* è un algoritmo che converte una stringa o un numero in un indice, cioè nel numero di record.

Tipicamente, il numero associato al record è molto più piccolo di ogni possibile valore che può assumere il numero originale.

Il metodo più comune di funzione HASH per *convertire un numero grande in uno più piccolo* è la divisione per un numero primo. Il numero di record è dato dal resto della divisione. Il numero primo deve essere almeno uguale alla lunghezza del file, cioè al numero di

Figura 4 – Listing del programma BUILD, che costituisce un vettore utilizzando la funzione HASH. Questo programma può essere adottato per operare sui files di un disco modificando opportunamente le istruzioni che riguardano gli accessi al vettore archivio.

```
100 REM     ==============================
110 REM
120 REM     COSTRUISCE IL FILE CON I NOMI ED I NUMERI
125 REM        TELEFONICI (PROGRAMMA BUILD)
130 REM
140 REM     ==============================
150 REM
160 LET     N = 11
170 DIM     N$ (10), F (N), S (N), X$ (1)
180 FOR     J1 = 1 TO N
190 LET     F (J1) = -1
200 NEXT    J1
210 LET     F1 = 1
220 REM
230 REM     ——— LEGGE IL NOME ———
240 REM
250 PRINT   "INTRODUCI IL NOME:";
260 INPUT   N$
270 IF      N$ < >" " THEN 300
280 PRINT   "FILE COSTRUITO: FINE PROGR. BUILD"
290 STOP
300 REM
310 REM     ——— TRASFORMA NOME IN INDICE, R ———
320 REM
330 GOSUB   1000
340 IF      R > 0 THEN 370
350 PRINT   "ERRORE – FILE PIENO"
360 STOP
370 REM
380 REM     ——— INTRODUCI IL NUMERO TELEFONICO ———
390 REM
400 PRINT   "INTRODUCI IL NUMERO TELEFONICO:";
410 INPUT   A (R)
420 LET     F (R) = 1
430 LET     S(R) = I1
440 REM     ——— INTRODUCI UN ALTRO DATO ———
450 GO      TO 230
```

Figura 5 – Listing del programma LOOK-UP, che ricerca un record nel vettore costruito da BUILD, utilizzando la funzione HASH. Questo programma può essere adottato per operare sul files di un disco modificando opportunamente le istruzioni che riguardano gli accessi al vettore archivio.

```
480 REM
500 REM     ==============================
510 REM
520 REM     RICERCA DI UN NUMERO TELEFONICO PER
525 REM     MEZZO DEL NOME (PROGRAMMA LOOK-UP)
530 REM
540 REM     ==============================
550 REM
560 PRINT   "INTRODUCI IL NOME:";
570 INPUT   N$
580 IF      N$< >""THEN 610
590 PRINT   "FINE DEL PROGRAMMA LOOK-UP"
600 STOP
610 REM
620 REM     ——— CHIAMA LA ROUTINE HASH PER
            TROVARE R ———
630 REM
640 LET     F1 = -1
650 GOSUB   1000
660 IF      R > 0 THEN 700
670 PRINT   "NOME NON TROVATO! PROVA CON  ALTRO"
680 GO TO    560
690 REM
700 REM     ——— STAMPA IL RECORD TROVATO ———
710 REM
720 PRINT   USING 730 : N$, A (R)
730 IMAGE   "NOME», 10A, «NUMERO DI TELEFONO", 6D
740 REM
750 REM     ——— RIPETI CON UN ALTRO NOME ———
760 REM
770 GO TO   560
780 REM
```

records che devono venire inseriti. È possibile calcolare sia il quoziente che il resto di una divisione nel modo seguente:

```
150 LET Q = INT (N1/N)
160 LET R = INT (N1-Q*N)
```

L'ordine in cui vengono eseguite le due istruzioni precedenti è molto importante. Il quoziente Q si calcola facilmente eliminando i decimali che risultano dalla divisione per N1.

Supponiamo che N1 = 25; ed N = 11; allora l'istruzione 150 dà come quoziente Q = 2; il valore di N*Q è 22, perciò quando si esegue l'istruzione 160 si ottiene R = 3.

È importante ricordare che R varia tra i valori 0 e N-1; quindi:

$$1 \leqslant R + 1 \leqslant N$$

e tutti i resti prodotti dalla divisione cadono nel campo dei numeri più piccoli. L'indice di valore R è esattamente ciò di cui abbiamo bisogno per accedere direttamente all'R-esimo record del file che contiene N records.

Sfortunatamente, spesso i valori di R prodotti in questo modo non sono unici, cioè ci può essere più di un numero maggiore di N1 che produce lo stesso valore di R. Per esempio, se N = 11, possiamo ottenere il valore R = 0 con N1 = 11, N1 = 22, ... In questo caso occorre calcolare un ulteriore numero per individuare univocamente un record.

Il modo più comune consiste nell'utilizzare il quoziente Q: un secondo tentativo verrà fatto sul settore R + Q, e, se anche questo settore non dovesse essere disponibile, si farà un ulteriore tentativo ad R + 2Q, e così via.

Nel caso in cui si stia effettuando una *ricerca* e non un inserimento, si controlla se il record in posizione R ha la chiave di ricerca voluta: in caso positivo si è ottenuto il record voluto, diversamente si prova col record in posizione R + Q, e così di seguito.

Si è così risolto il problema di ridurre un numero grande ad un numero piccolo, ma resta ancora da trovare il modo di *convertire un nome in un numero R.*

Le funzioni BASIC ASC e SEG sono utili a questo scopo.

La funzione ASC converte una lettera dell'alfabeto in un numero intero compreso tra 65 e 92. La funzione SEG estrae una lettera da una stringa, quando è usata nel modo seguente:

```
310 LET X$ = SEG (N$, J, 1)
```

In questo caso, la funzione SEG estrae la lettera in posizione J da N$ e la memorizza in X$.

Possiamo usare questa funzione più volte, facendo variare J da 1 alla lunghezza di N$, in modo da estrarre tutte le lettere di N$ una dopo l'altra.

Unendo l'algoritmo di conversione alfabetica alla divisione per un numero primo, si ottiene il sottoprogramma seguente:

```
1000 REM  ==============================
1010 REM
1020 REM                FUNZIONE  HASH
1030 REM
1040 REM  ==============================
1050 REM
1060 REM  N$=NOME CONOSCIUTO DELLA CHIAVE
1070 REM  I1 =CHIAVE ESPRESSA COME NUMERO
1080 REM  N =LUNGHEZZA DELLA TABELLA E DIVISORE
1090 REM  F =VETTORE DEI FLAGS PER GLI INDICI
1100 REM      DOVE (-1) SIGNIFICA VUOTO
1110 REM      E (+1) SIGNIFICA PIENO
1120 REM  F1 =CODICE DI OPERAZIONE, (-1)
1130 REM  SIGNIFICA RICERCA E (+1) SIGNIFICA
1135 REM  INSERIMENTO INIZIALE
1140 REM  R  =INDICE CALCOLATO DAL
          SOTTOPROGRAMMA
1150 REM  S  =VETTORE DATI, CONTIENE I1
1160 REM
1170 REM  --- CALCOLO DI I1 ---
1190 LET  I1 =0
1200 FOR  J =1 TO LEN (N$)
1210 LET  X$=SEG (N$, J, 1)
1220 LET  X=ASC (X$) -65
1230 LET  I1 =I1 +J*X
1240 NEXT J
1260 REM  ---APPLICA FUNZIONE HASH ---
1270 REM
1280 LET  Q1 =INT (I1/N)
1290 LET  R1 =INT (I1 - Q1 *N)
1300 LET  R =R1 +1
1310 LET  Q=Q1 +1
1320 REM  --- RIPETE I CONFRONTI ---
1330 FOR  J1 =1 TO N-1
1340 IF   F(R) >0 THEN 1380
1350 IF   F1 >0 THEN 1480
1360 LET  R =0
1370 RETURN
1380 REM  ---CONFRONTO CON LA CHIAVE---
1390 IF   S (R) =I1 THEN 1480
1400 LET  R =R +Q1
1410 LET  Q +INT (R/N)
1420 LET  R1 =INT (R-Q*N)
1430 LET  R=R1 +1
1440 NEXT J1
1450 REM  --- RIENTRO PER ERRORE---
1460 LET  R=0
1470 REM  --- RIENTRO NORMALE---
1480 RETURN
```

Si noti come il valore che viene ritornato al posto di ogni lettera di N$ viene pesato con J e sommato per ottenere I1. Questo metodo fornisce *quasi sempre* un valore unico di I1 al posto di un nome.

La funzione HASH usa tre vettori per ogni record. Il valore F si riferisce alla disponibilità di un settore del file; se F (R) = -1 il record (il settore) è libero e quindi utilizzabile, se F (R) = + 1 il record è già stato occupato. Il valore di I1 per ciascun nome è memorizzato nel vettore S; questa condizione è necessaria per garantire la possibilità di confronto tra I1 e S durante la ricerca. Il dato (in questo caso il numero telefonico) è memorizzato in un terzo vettore chiamato A che non compare nel sottoprogramma.

La routine HASH viene usata per costruire un file quando F1 = 1 e per cercare un record in un file quando F1 = -1; così, l'utilizzo di F1 permette di usare la stessa routine sia per l'inserimento nel file che per la ricerca sul file stesso.

Le routines di costruzione del file e di ricerca nel file sono notevolmente semplificate quando si usa il metodo HASH. Si può passare un nome alla routine (mediante il vettore N$) e, quando F1 = 1, siamo nelle condizioni di inserire un nuovo nome con numero di telefono nel file, quando F1 = -1, invece, passiamo un nome alla routine per ottenere come risposta il numero telefonico associato a questo nome in un inserimento precedente. ■

Bibliografia:

«The Mind Appliance: Home Computer Application» T.G. Lewis - Hayden Book Company, Inc.

Nota

(¹) Ciascuna voce che compare in parentesi indica una forma generale di un dato argomento. Esempio: [variabile] = una qualunque variabile; [identificatore di linea] = un qualunque identificatore di linea.

PROGRAMMAZIONE
STRUTTURATA

Introduzione alla programmazione strutturata

di **Chip Weems** - *Oregon State University*

La programmazione strutturata è un insieme di tecniche di progetto per razionalizzare lo sviluppo del software e per ridurre tutti quegli effetti che distolgono in modo non accettabile il programmatore dalla effettiva programmazione. L'uso di linguaggi strutturati come il PASCAL promuove delle buone tecniche di programmazione.

Comprare un sistema computerizzato, quando l'industria degli elaboratori era al suo inizio, significava pagare soprattutto la macchina. La parte più costosa della macchina stessa era il banco di memoria: 16 K bytes di memoria erano considerati molti e 64 K potevano essere trovati solo su dei super-elaboratori! Tutto questo significava che i programmi dovevano, innanzitutto, sfruttare il poco spazio disponibile.

Per fortuna l'hardware si è evoluto ed è divenuto meno caro. Grandi memorie ed alte velocità sono, ora, caratteristiche comuni a molti sistemi. Di colpo ci si è accorti che la velocità e lo spazio non sono così importanti. In effetti i ruoli si sono invertiti e l'hardware è la parte meno cara. Vedi figura 1.

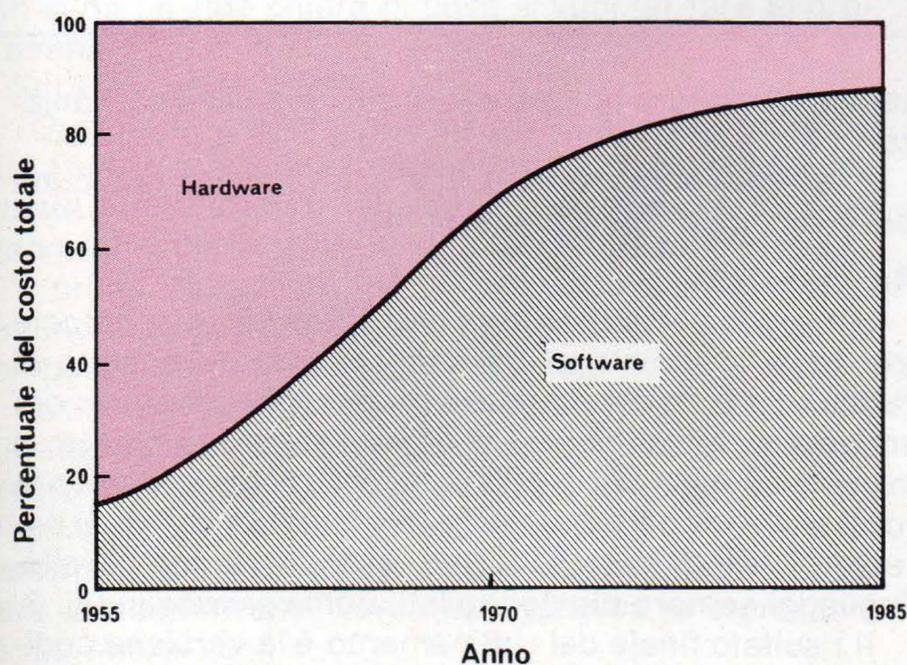



Figura 1 - Tendenza del costo Hardware/Software.

La parte più costosa per chi «oggi» possiede un elaboratore, è il programmarlo. Con l'avvento di un hardware meno caro, va ora messo a fuoco non più il rapporto velocità-spazio, ma il rapporto costo di programmazione-costo della macchina. Il nuovo obbiettivo è di sfruttare maggiormente il tempo di programmazione: l'efficienza del programma è divenuta meno importante - è più facile aggiungere altro hardware.

Ci sono alcune importanti osservazioni che dovrebbero essere fatte riguardo alla moderna programmazione. Anzitutto bisogna notare che il costo maggiore di un centro non è più nello sviluppo iniziale ma nella manutenzione e revisione. Ad esempio un programma medio che richiede tre mesi uomo per la sua messa a punto può avere una vita di dieci anni e più, durante la quale possono essere necessarie dozzine di cambiamenti. Questo può aggiungere parecchi anni di lavoro.

È anche interessante notare che la maggior parte del tempo speso nella revisione di un programma è dedicata all'analisi del codice esistente. Questo è il tempo necessario ad un programmatore per spezzare il programma in blocchi di codice. Anche nella fase di sviluppo, la maggior parte del tempo non è, solitamente, dedicata alla progettazione ed alla codifica, ma alla ricerca degli errori. Il tempo che il programmatore dedica alla programmazione è piccolo rispetto a quello dedicato a lavori di contorno (fig. 2).

Sfortunatamente, le tecniche software non si sono sviluppate come l'hardware, dopo la comparsa dei primi linguaggi ad alto livello. La dimostrazione è che due dei più popolari linguaggi in uso oggi, il FORTRAN ed il COBOL, sono dei relitti degli ultimi anni 50.

Ecco una grossa contraddizione nell'industria dei computers: hardware della 4ª generazione contrapposto ad un software, come il BASIC o editors orientati alla linea, basati su progetti vecchi di 15 anni e fatti su un hardware della 2ª generazione.

La programmazione strutturata è un tentativo di modernizzare lo sviluppo del software e ridurre gli effetti spuri che sottraggono tanto tempo alla programmazione vera e propria. La p.s. spinge ad enfatizzare maggiormente la fase di analisi per ridurre il tempo di correzione e per migliorare la organizzazione dei programmi. Inoltre delle speciali tecniche di codifica rendono più facile la revisione. L'uso di linguaggi strutturati, come il PASCAL, rendono i programmi più affidabili permettendo ai compilatori di inserire un numero maggiore di controlli nei programmi che generano.

Tecniche strutturate

La p.s. è un insieme di tecniche di codifica e progetto che guida gli sforzi del programmatore nella produzione di programmi organizzati, ben strutturati ed affidabili. Spesso si identifica la p.s. con il metodo top-down. Questa identificazione è fuorviante poichè sono usati sia il metodo top-down che il metodo bottom-up. Le tecniche *top-down* si riferiscono a metodi che partono dal problema nella sua globalità ed attraverso stadi successivi lo suddividono in problemi sempre più piccoli e più facilmente trattabili. Le tecniche *bottom-up* usano l'approccio opposto: partendo da un livello di dettaglio, concatenano piccole unità per formarne di più grandi e ripetono il processo fino all'ottenimento della soluzione dell'intero problema.

Le tecniche strutturate presuppongono la divisione della fase di sviluppo di programmi in due: progetto e implementazione. Durante la fase di progetto, che è quella su cui si pone maggiormente l'accento, si usa quasi esclusivamente il metodo top-down. Nella fase di implementazione, tuttavia, si privilegia un approccio di tipo bottom-up.

Figura 2 - Modalità di sviluppo di un sistema.

Risorse uomo

Proposta

Trattativa contratto

Pianificazione

Esame progetto di massima

Progetto sistema

Progetto dettagliato

Test codice

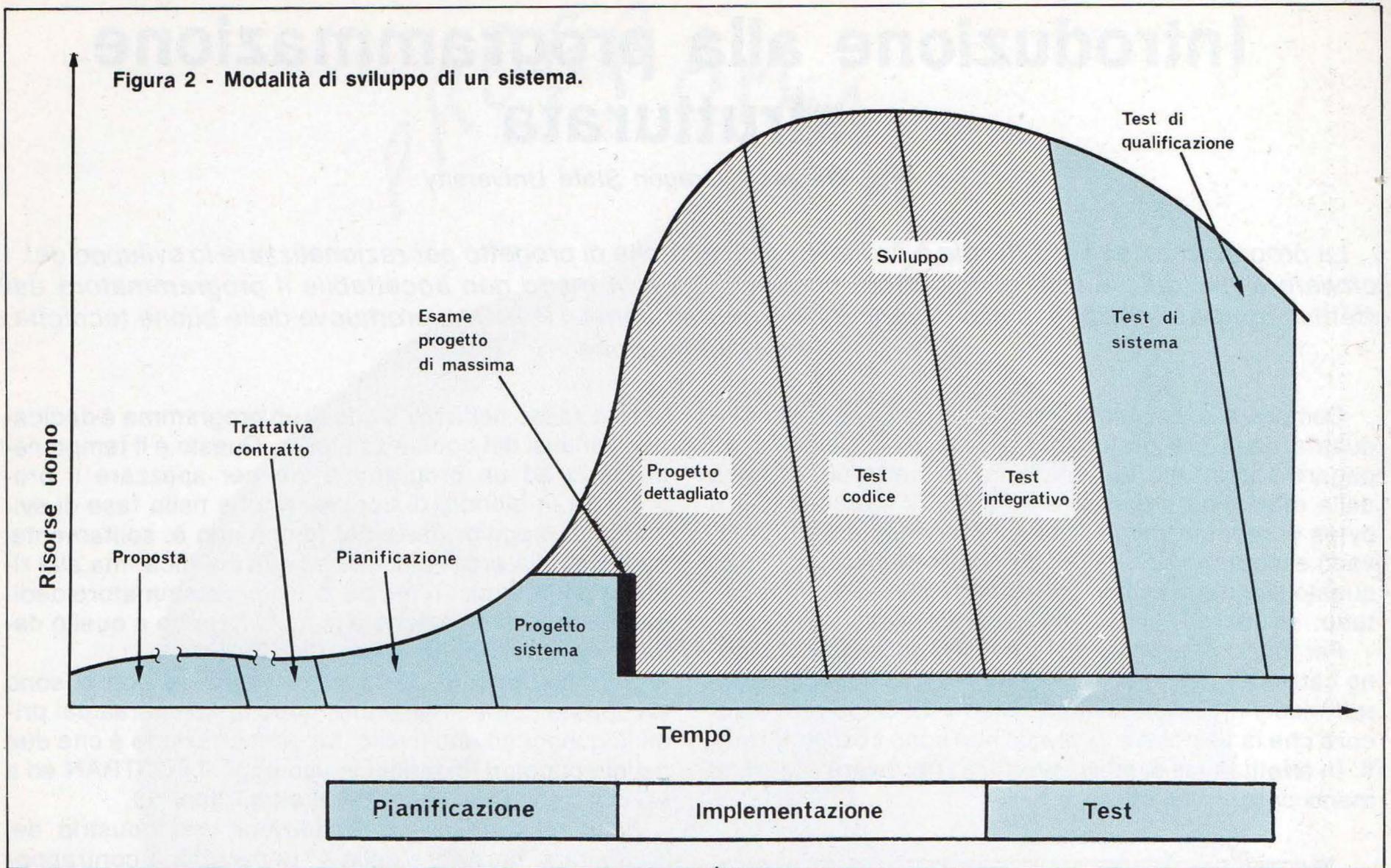
Sviluppo

Test integrativo

Test di qualificazione

Test di sistema

Tempo

Pianificazione	Implementazione	Test

L'analisi di qualunque problema è di natura tale da dover essere più logicamente affrontata in top-down.

La nostra prima regola di p.s. è perciò: si inizi analizzando il problema in modo top-down; lo si spezzi in blocchi più piccoli in modo che l'intera organizzazione rimanga visibile.

Questa è la filosofia guida dell'intero processo di progettazione il cui risultato finale è un programma correttamente codificato. Le tecniche più importanti, usate qui, sono i raffinamenti successivi e la scomposizione in blocchi. È da notare che, praticamente, queste non sono mai pure top-down. Un programmatore esperto sa cosa è possibile e cosa no; un tale programmatore sa guardare avanti in modo da evitare dei progetti impossibili ad attuarsi. Mentre il progetto è in sviluppo devono, costantemente, essere considerati dei possibili schemi di codifica.

Usata in modo corretto questa tecnica può essere una preziosa valutazione dei mezzi e può aumentare in modo significativo la velocità di progettazione. Il punto da tener presente è che il programmatore non dovrebbe mai essere talmente preso dalla codifica da abbandonare completamente l'approccio top-down.

L'implementazione è fatta meglio con le tecniche bottom-up.

Terminato il progetto, è implementato e provato per primo il modulo indipendente, a livello più basso. Le routines a livello più alto sono costruite usando queste subroutines fino a che è costruito il programma ed ha luogo la verifica finale.

Sorprendentemente un progetto accurato ed una implementazione in questa forma non necessita di un tempo apprezzabilmente più lungo dei vecchi metodi. Sono certamente necessari certi vincoli per il programmatore ma, visti i risultati, è difficile capire perchè si voglia continuare ad usare i vecchi metodi non organizzati.

A questo punto è possibile approfondire qualcuna delle tecniche menzionate prima.

Raffinamento

Il metodo dei *raffinamenti successivi* è un approccio top-down alla sintassi del programma. Il concetto di raffinamento fa riferimento al linguaggio usato per descrivere la soluzione che deve essere programmata: si inizia con frasi molto generali su cosa il programma o la routine dovrebbero fare, si lavora quindi, attraverso parecchi livelli di specificazione, usando, ad ogni livello, linguaggi sempre più dettagliati, come si vede in fig. 3.

Il risultato finale del raffinamento è la versione codificata del programma o del modulo.

Per poter applicare i raffinamenti, è necessario avere preventivamente specificato - almeno in parte - come il programma può essere spezzato da un punto di vista funzionale. Un'altra tecnica, tecnica delle *scomposizioni successive*, guida l'analista nel determinare come il programma debba essere frazionato. In generale raffinamenti e scomposizioni vanno in parallelo, con queste ultime in leggero anticipo.

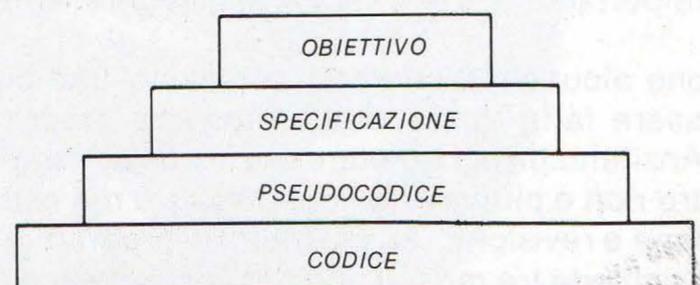



Figura 3 — Raffinamenti successivi. Si inizia con la scrittura dell'obiettivo, con pochi dettagli. Andando verso i livelli più bassi si entra sempre più nel dettaglio, fino a che, al quarto livello, si ha un programma codificato.

L'obbiettivo dei raffinamenti successivi è di ridurre la complessità dello sviluppo dei programmi. Questo è organizzato in una sequenza di livelli di complessità stabile durante il progetto: raggiunto un nuovo livello, si inizia a sviluppare quello che segue. Questa tecnica assomiglia al modo in cui uno scalatore raggiunge una vetta: viene stabilita una sequenza di campi base, ciascuno dei quali fornisce sia il punto di partenza per la scalata del giorno successivo, sia la salvezza nel caso di fallimento.

Sorprendentemente parecchi programmatori attaccano ancora programmi «montagna» con un unico sforzo: ci sono troppe informazioni in un grosso programma, e la mente umana non può considerare tutto in una volta. Anche se il programma risultante gira, il suo autore, probabilmente, non riesce a spiegare tutto e a garantirlo senza errori.

Sviluppo di un programma: obbiettivo

Quanti livelli sono necessari nello sviluppo di un programma? Dipende dalla sua dimensione e complessità, ma in genere è richiesto un minimo di quattro livelli: *obbiettivo, specificazione, pseudocodice, codice.*

La prima cosa da fare è scrivere, in chiaro, l'obbiettivo che si ha in mente. Poi occorre esaminare attentamente se la formulazione è sufficientemente esplicita oppure poco chiara, ecc. Se non si riesce a rendere esplicito l'obbiettivo tramite la scrittura, certamente non si ha un'idea chiara di cosa si vuol far fare al programma.

Scrivere ad esempio:

«Questo programma servirà per quadrare i conti del mio libretto di banca» è notevolmente diverso dalla seguente formulazione: *«Questo programma servirà per elaborare tutte le entrate e le uscite della famiglia».*

Il primo obbiettivo, infatti, potrebbe essere scritto per una calcolatrice programmabile, mentre il secondo comporta lo strutturare una completa base di dati.

Invece un esempio di obbiettivo scritto chiaramente è il seguente, relativo al gioco della roulette: *«Questo programma simulerà il gioco della roulette come viene giocato con le regole di Montecarlo. Permetterà un massimo di cinque giocatori alla volta e, per semplificare le scommesse, userà uno schema di ingresso a forma libera».*

Nella prima frase è stato specificato l'obbiettivo principale e certi punti secondari. Si noti che l'obbiettivo è abbastanza generale, in quanto si suppone che i dettagli vengano introdotti solo a livello di specificazioni.

Specificazioni

Puntualizzato in modo soddisfacente l'obbiettivo, si tratta di affrontare le specificazioni, che si possono dividere in tre parti: *ingresso, uscita, strategia.*

Le specificazioni permettono di scrivere delle idee, per ora abbozzate, su come sarà l'elaborazione.

La tabella I è uno dei possibili esempi del programma della roulette.

Ovviamente questa non è una descrizione completa dell'ingresso/uscita, ma è ragionevolmente buona. Se si stesse effettivamente scrivendo questo programma, le specificazioni sarebbero più dettagliate e sarebbero inclusi parecchi esempi. Frasi che possono comparire nella descrizione di un ingresso/uscita sono: tipo di schede, formato dei flussi, descrizione dei records... Se questa fosse una subroutine, le specificazioni descriverebbero anche i parametri passati e ritornati al programma principale.

Tabella I — Questo è l'abbozzo del gioco della roulette che sarà usato per sviluppare il programma. L'abbozzo definisce l'ingresso e l'uscita e le azioni principali del programma. Dovrebbe essere veramente generale e dare una visione globale del progetto.

Ingresso:
- Numero dei giocatori : intero 1/15 inclusi
- Nomi dei giocatori: stringa troncata a 15 caratteri
- Risposte si/no: stringa troncata ad un carattere
- Poste: stringa troncata a 72 caratteri della forma:
  - numero intero valore della posta in gioco **su** parola chiave di posta. Tutte le parole e simboli estranei sono ignorati. Invece della posta, si possono immettere le parole chiave speciali **abbandono** e **passo.**
  - Parole chiave di posta: numeri separati da virgole.
  - 12H, 12M, 12L (dozzine)
  - 12A, 12B, 12C (colonne)
  - basso, alto (metà)
  - pari, dispari
  - rosso, nero

Uscita:
- Domanda del numero dei giocatori.
- Richiesta dei nomi dei giocatori.
- Domanda se necessitano istruzioni.
- Istruzioni.
- Richieste delle poste.
- Risultati e frasi di vincita.
- Messaggi di poste errate.
- Messaggi di saluto.

Strategia:
- Dopo la sequenza di partenza loop per l'ingresso delle poste.
- Elabora ogni posta facendo la scansione della stringa ricercando abbandono, passo o una cifra.
- Posiziona il segnale di abbandono o passo se è stato trovato uno dei due e passa alla prossima posta. Se è una cifra continua a leggere le cifre e converti in decimale fino al **su,** quindi scandisci la parola chiave. Finito di elaborare la posta, verificane la validità. O accetta la posta o stampa un messaggio di errore. Scegli il numero casuale vincente, determina chi ha perso e chi ha vinto, stampa i risultati, verifica i segnali di passo e abbandono per azioni speciali. Riparti con il loop iniziale se ci sono ancora dei giocatori, altrimenti saluta e fermati.

A questo punto, secondo il livello del processo di raffinamento, è importante la strategia: *non ci devono però essere frasi dipendenti dal sistema o dal linguaggio.* Dalla descrizione di un programma a livello di specificazioni bisognerebbe riuscire a strutturare qualsiasi task e ad implementarlo su qualunque sistema, con sufficienti risorse, senza dover cambiare le specifiche.

Di nuovo occorre esaminare ciò che è stato fatto. La strategia è troppo o troppo poco dettagliata, dipende da caratteristiche del sistema o del linguaggio, è troppo complessa? L'ultima domanda è molto importante: se la risposta è affermativa, significa che il problema non è stato correttamente scomposto. *Bisogna allora ritornare alla scomposizione per verificare come è stata fatta.*

Pseudocodice

Nell'ulteriore fase di raffinamento bisogna essere più specifici. È a questo punto che possono apparire altri livelli di raffinamento su grossi programmi. Comunque un programma progettato bene ha raramente bisogno di più di 4 livelli; se richiede livelli multipli di pseudocodice, è stato probabilmente scomposto in modo non corretto. A questo livello si possono aggiungere caratteristiche che dipendono dal sistema, ma bisogna evitare di legarsi al linguaggio di programmazione. Lo

LISTING I: Pseudocodice per il programma della roulette. Questo livello di progetto è usato per una prima determinazione di cosa il programma dovrebbe fare. Non prende in considerazione gli aspetti del livello basso del programma.

```
Inizio programma.
  Chiedi quanti giocatori.
  Per quanti giocatori ci sono,
    Acquisisci il nome di ognuno.
  Chiedi se sono necessarie istruzioni.
  Se si, stampa le istruzioni.
  Finchè ci sono dei giocatori lasciati fuori.
    Per quanti giocatori ci sono,
      Ripeti fino a che è ottenuta una posta valida:
        Acquisisci la posta dei giocatori.
        Fa la scansione della posta.
        Verifica la validità della posta.
    Determina il numero vincente.
    Per quanti giocatori ci sono,
      Se il giocatore abbandona, elabora il saldo
      Se il giocatore è passato, elabora il passo
      Se il giocatore punta,
        Determina se ha vinto o perso.
        Elabora in accordo con sopra.
Fine programma.
```

pseudocodice dovrebbe quindi usare liberamente un'ampia gamma di frasi che si potranno poi spezzare nei costrutti che il linguaggio scelto effettivamente permette. Usare una grande varietà di costrutti abbastanza potenti permette di progettare più liberamente senza cadere nelle pastoie di un particolare linguaggio.

Spesso si usa scrivere gli psudocodici in una qualche forma di ALGOL in quanto la struttura a blocchi di questo linguaggio si presta molto bene per pseudocodici strutturati. Alle volte questo «ALGOL» assomiglia maggiormente a un «PASCAL»: le differenze non sono comunque significative in quanto i due linguaggi sono simili, ed un buon codificatore in pseudocodici tenta in ogni modo di eliminare le dipendenze dal tipo di linguaggio.

Lo pseudocodice per il gioco della roulette potrebbe essere qualcosa come il listing I. Questo è ancora abbastanza ad alto livello, ma è già più vicino ad un programma che non la strategia.

Usato correttamente, uno pseudocodice può sostituire un flow-chart: migliorando anche la facilità di comprensione.

Una regola che bisognerebbe sempre seguire nello scrivere uno pseudocodice è di renderlo comprensibile a chiunque, programmatori e non.

Una prova importante è: se un'altra persona può correttamente interpretare un vostro pseudocodice e, usandolo, fare la parte dell'elaboratore senza scontrarsi con grossi problemi (bachi), allora è scritto bene. Partendo dal listing I si può facilmente convertire in un linguaggio come PASCAL, PLI, ALGOL...

Codice

Passando a considerare i codici si può senz'altro affermare che ben raramente un programma principale dovrebbe superare 1 o 2 pagine. Se le supera, è probabilmente troppo complesso, e parte delle sue operazioni dovrebbero essere implementate come subroutines. Dividere il programma in spezzoni gestibili è l'idea che sta dietro le scomposizioni successive. Come è già stato fatto notare, la scomposizione è l'altra metà della fase di progetto. Ora che si è finito di considerare i raffinamenti successivi, vediamo più approfonditamente le scomposizioni.

Scomposizione

Le scomposizioni successive sono un approccio top-down alla semantica del programma. Usando la scomposizione si esamina il significato di un segmento di programma per determinare le sue funzioni nel complesso. Lo scopo è quello di determinare la complessità della funzione per vedere se può essere ulteriormente scomposta. La scomposizione fa riferimento allo *spezzettamento delle funzioni del programma* per ridurne la complessità e renderlo più gestibile.

Il risultato finale delle scomposizioni è che tutte le funzioni del programma sono ridotte ad un insieme di subroutines semplici ed isolate, come si vede in figura 4. Come per i raffinamenti, tuttavia, lo scopo delle scomposizioni è di fornire una sequenza di livelli, come i già citati «campi base», che riducano il processo di progettazione da un unico gigantesco balzo ad una serie di brevi e facili passi.

Per la scomposizione, diversamente dai raffinamenti, non esiste un numero *fisso* di livelli: sono solo la complessità e le dimensioni che determinano se un programma ha bisogno di 1 o 20 livelli. Con i raffinamenti si ha il vantaggio di lavorare sempre con moduli semplici e scomposti i quali non dovrebbero mai superare un certo grado di complessità. La scomposizione, invece, è usata per aprire nuove «piste» ed è applicata a programmi di complessità qualunque.

Dal momento che non ci sono livelli fissi di scomposizione, bisogna avere delle regole che specifichino quali sono le condizioni per creare un nuovo livello. Non essendo la programmazione una scienza esatta, le regole che si proporranno sono di tipo euristico: delle tracce che guidano nella fase di progettazione e non delle leggi a cui obbedire.

Si prenderanno in considerazione due importanti caratteristiche dei moduli che serviranno come base per la proposta di queste regole. Esse sono: *grado di connessione* e *relazione funzionale*.

Il grado di connessione fa riferimento al numero di canali di comunicazione tra un modulo ed il resto del programma.

Un modulo con un elevato grado di connessione ha molti legami con le altre parti del programma; uno con bassa connessione ha pochi legami.

Il livello di relazione funzionale interno è una descrizione di come sono collegati fra di loro i processi di un modulo. Un'alta correlazione tra i processi sta a significare che tutti sono coinvolti nel fornire la funzione globale del modulo. Un gruppo di processi ha una bassa relazione funzionale nei riguardi del modulo se non

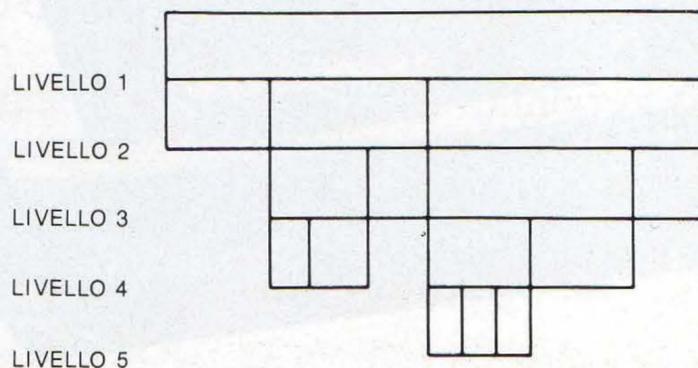



Figura 4 — Scomposizione successiva. Si inizia spezzando l'intero problema in sottoproblemi più piccoli. Questo processo è ripetuto fino a che si sono ottenuti i moduli indipendenti. Non per tutti i moduli servirà la stessa profondità di scomposizione.

contribuisce in modo sostanziale all'espletamento delle funzioni del modulo.

La misura del livello di interrelazione all'interno di un modulo è detta coesione.

Ora che abbiamo definiti i termini, consideriamo un po' più approfonditamente queste due caratteristiche e vediamo come sono legate alla p.s.

Connessioni tra moduli

I programmi i cui moduli sono collegati da un gran numero di canali di comunicazione sono ovviamente più complessi di quelli con pochi legami. È difficile lavorare con programmi i cui moduli sono pesantemente collegati, come in fig. 5, in quanto capire un modulo significa capire anche tutti gli altri che con esso sono collegati. Questo può innescare una reazione a catena che richiede al programmatore di progettare l'intero programma fino ad un livello di dettaglio tale, che il compito diviene impossibile.

Un altro aspetto negativo di un alto grado di connessione è che ogni cambiamento in un modulo ha effetti, difficilmente controllabili, su tutto il programma: gli errori si propagano con estrema facilità da un modulo all'altro. Poichè uno degli obiettivi della p.s. è di ridurre la complessità ed il numero di errori, gli sforzi di progettazione dovrebbero rivolgersi a ridurre il numero di connessioni tra i moduli, come in fig. 6.

I moduli sono generalmente di due generi: subroutines o blocchi di codice dentro un programma. Essi possono comunicare tramite aree in comune, variabili condivise, parametri formali e flusso del controllo. (Le aree dati in comune producono programmi altamente connessi come in fig. 7). Si possono dividere le informazioni che transitano lungo questi percorsi in due tipi: quelle che influenzano i dati e quelle che agiscono sul flusso del controllo. Anche i percorsi possono essere di due generi: quelli che connettono ad un modulo e quelli

Figura 5 — Moduli connessi in modo forte. I larghi canali di comunicazione non permettono di separare le funzioni in moduli. Inoltre, errori dovuti ad un cambiamento possono propagarsi in tutto il programma.

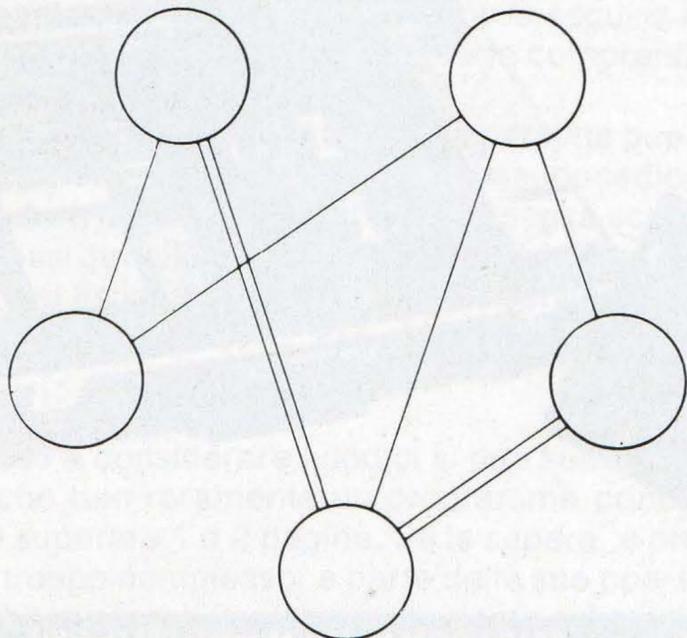



Figura 6 — Moduli connessi in modo debole. Linee di comunicazioni strette esaltano le qualità di «scatole nere» dei moduli. Servono anche per isolare gli errori.

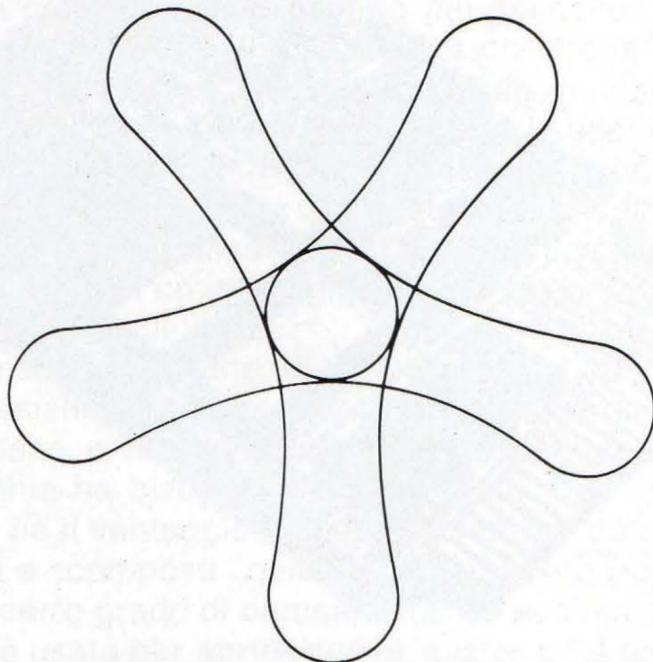



Figura 7 — Effetti di aree dati in comune. Ciascun modulo è fortemente connesso ad ogni altro che accede ai dati: in pratica non c'è isolamento. Ogni azione di qualsivoglia modulo può perciò avere effetti non trascurabili sugli altri. Si noti la somiglianza con i moduli connessi in modo forte.

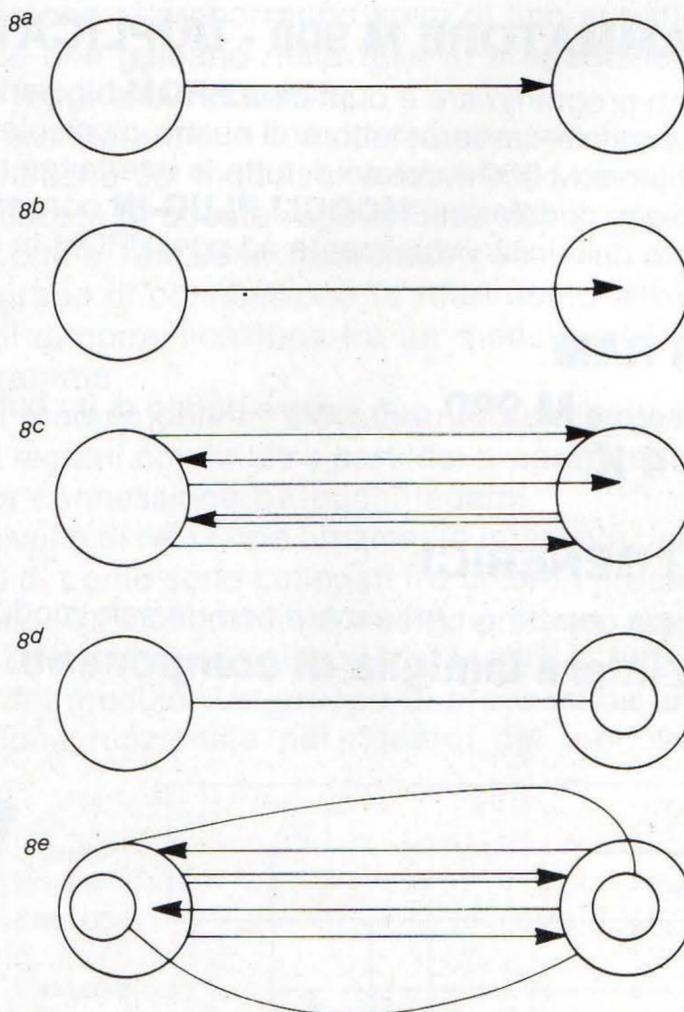



Figura 8 — Tipi di comunicazione. La figura (a) mostra la connessione semplice esterna, che è la più desiderabile. In (b) si ha una connessione semplice interna. La figura (c) è una comunicazione complessa. In (d) si vede una comunicazione in cui un modulo modifica l'interno di un altro. Per ultima, la figura (e) mostra il tipo peggiore di comunicazione: una combinazione complessa di tutte le altre forme.

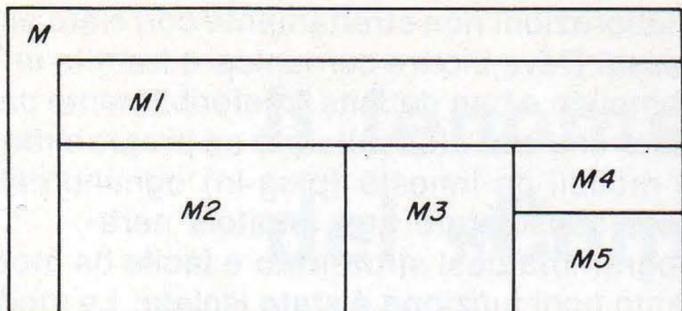



Figura 9 — Classi di relazioni. La forma più forte di coesione è la relazione funzionale in cui ogni processo è implicato totalmente nell'espletare la funzione globale del modulo.

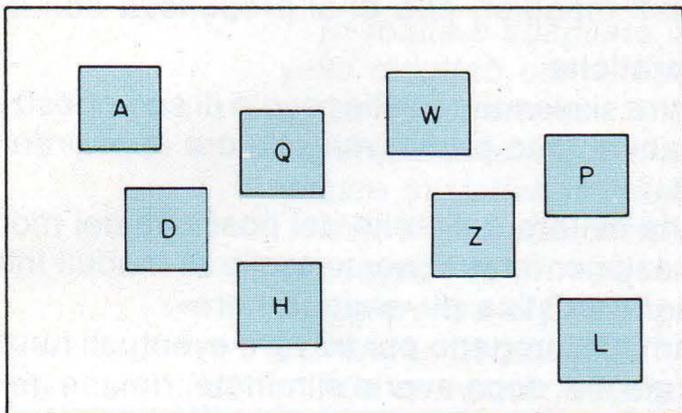



Figura 10 — Classi di relazioni. Qui è mostrata la relazione casuale, in cui i processi sono incorrelati se non per il fatto di trovarsi nello stesso modulo.

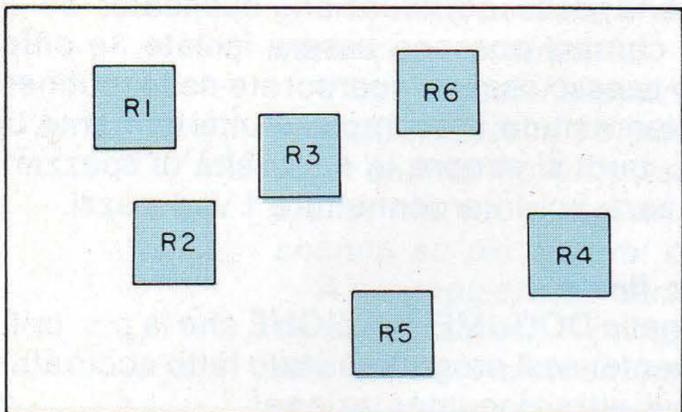



Figura 11 — Classi di relazioni. Qui si vede la relazione di tipo logico, nella quale i processi sono raggruppati in quanto svolgono funzioni simili.

che connettono con qualcosa dentro il modulo. La fig. 8 illustra i vari mezzi di comunicazione.

Il grado di connessione per un percorso è determinato dalla complessità delle informazioni comunicate, dal tipo di informazioni e dal tipo di percorso. Il grado di connessione è quasi impossibile da valutare quantitativamente, ma, qualitativamente, si può vedere che certi tipi di connessione sono migliori o peggiori di altri.

Con maggior precisione diciamo che: le connessioni semplici sono migliori di quelle complesse, i percorsi che portano al modulo sono preferibili a quelli che conducono all'interno del modulo, le comunicazioni che influenzano i dati sono preferibili a quelle che agiscono sul flusso del controllo. Di gran lunga la peggior forma di connessione è quella che modifica il codice interno di un altro modulo, perchè richiede che il modulo modificante «sappia» come l'altro modulo lavora a livello di codice macchina. Perciò ogni cambiamento del modulo modificato richiede un cambiamento nel modificante.

Connessioni esterne sono preferibili a quelle verso l'interno, in quanto in ogni istante l'intero modulo dovrebbe poter gestire ogni comunicazione. Le connessioni interne possono bypassare certi controlli e fanno sì che il modulo non possa controllare il suo stato interno.

Le comunicazioni che agiscono sul flusso del controllo richiedono che il modulo conosca come il modulo ricevente lavora: per questo la comunicazione di soli dati è preferibile.

I dati possono a loro volta essere divisi in due tipi: dati di grande estensione e dati di piccola estensione. Quelli di grande estensione includono variabili globali, aree di dati comuni, dati condivisi come in fig. 7. Come dati di piccola estensione si possono considerare i parametri formali passati tra subroutines, i quali producono connessioni meno complesse in quanto sono molto specifici.

Un cambiamento fatto su un'area di dati comuni porta a riconsiderare tutti i moduli connessi; esiste inoltre il problema di determinare quali moduli possano modificare quei dati. Che politica si adotta nell'accesso ai dati? Tutti possono accedere direttamente o l'accesso avviene solo tramite un modulo di interfaccia...? Per queste ed altre ragioni, trattare con dati in aree comuni può essere problematico.

Questa dovrebbe essere la configurazione migliore: *tutti i moduli comunicano passando una quantità minima di dati semplici, attraverso parametri formali ben definiti.* Naturalmente questo richiede che il programma sia stato spezzato in moduli che svolgono un'unica semplice funzione. Si analizza ora la coesione.

Relazioni all'interno dei moduli

Le relazioni che i processi interni ad un modulo possono avere tra di loro o con il modulo stesso sono state raggruppate in sei classi: qui sotto sono ordinate partendo dalla relazione più debole.

— Casuali
— Logiche
— Temporali
— Di comunicazione
— Sequenziali
— Funzionali

Dalla discussione sul grado di coesione si vede che è altamente desiderabile comporre moduli tali che tutti i processi interni siano direttamente messi in relazione con la specifica funzione del modulo: *funzionalmente* correlati, come mostrato in fig. 9. Per quanto riguarda la coesione, quindi, il livello più alto è il migliore. Bisognerebbe cercare di progettare dei moduli tali che i loro processi interni siano collegati nella maniera più forte possibile. Si esaminano ora le classi di relazione più deboli.

I processi *casualmente* collegati, mostrati in fig. 10, sono completamente incorrelati se non per il fatto di risiedere nello stesso modulo. Questa è la forma più debole di relazione; ogni modulo così composto dovrebbe essere diviso in moduli separati: uno per ogni processo.

I processi collegati *logicamente,* mostrati in fig. 11, non hanno nessuna relazione reale se non per il fatto che svolgono funzioni simili, e perciò sono stati raggruppati per formare un modulo; anche questo andrebbe spezzato in moduli più piccoli.

I processi che sono *temporalmente* collegati, come in fig. 12, sono identici al caso precedente con l'ulteriore vincolo dell'esecuzione nello stesso tempo. Caratteristica è la routine di inizializzazione che c'è in quasi tutti i programmi. Molto di ciò che solitamente viene

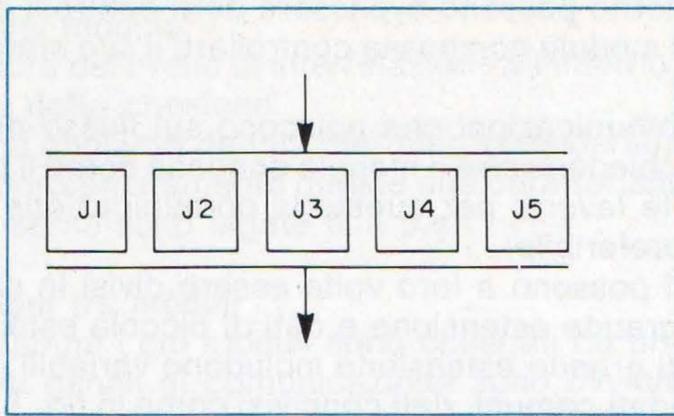



Figura 12 – Classi di relazioni. In una relazione temporale i processi possono svolgersi in qualunque ordine: parallelismo in senso lato.

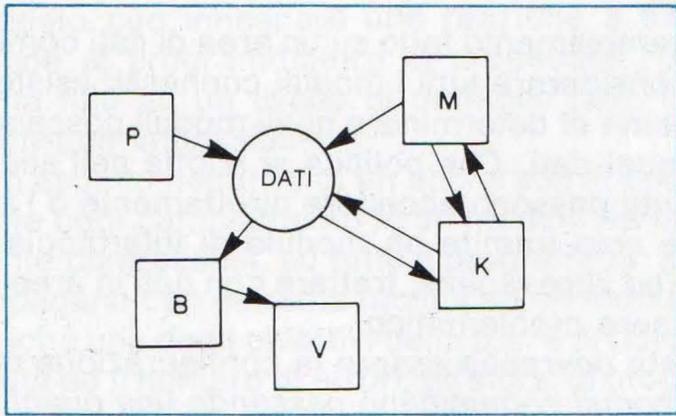



Figura 13 – Classi di relazioni. I processi comunicano direttamente tramite dei dati in comune.

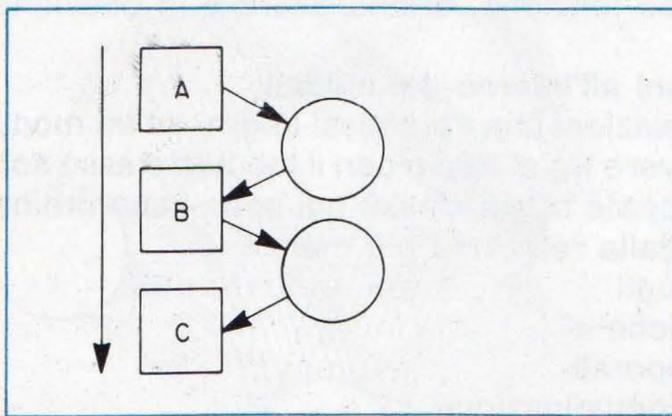



Figura 14 – Classi di relazioni. Relazione sequenziale in cui un processo genera i dati richiesti come ingresso dal successivo.

chiamato inizializzazione potrebbe essere fatto più tardi nel programma e potrebbe essere distribuito tra i moduli.

I processi con relazioni *di comunicazione* sono quelli raggruppati in un unico modulo perchè si suddividono dei dati come si vede in fig. 13. Questa è una relazione più forte, ma bisognerebbe tentare di suddividere anche questo tipo di modulo. Una soluzione abbastanza diffusa è quella di implementare un modulo master che gestisce i dati e li passa a dei sottomoduli quando sono loro necessari.

I processi collegati sequenzialmente (fig. 14) sono tali che ognuno crea i dati per quello che segue. Naturalmente un modulo di questo genere può essere suddiviso in più sottomoduli, ognuno dei quali è chiamato in sequenza da un altro modulo «principale» che, semplicemente, gestisce il passaggio dei parametri e del controllo. Questo ha il vantaggio di rendere questi processi elementari direttamente accessibili al resto del programma.

Poche connessioni con gli altri moduli ed un'alta coesione interna: queste sono le caratteristiche ideali di un modulo. Per ottenere ciò, ogni modulo deve espletare una funzione ben specificata e non deve contenere elaborazioni non strettamente correlate alla funzione stessa. Deve inoltre comunicare tramite un'interfaccia semplice e ben definita, preferibilmente passando parametri formali. Il risultato è un programma composto di moduli da innesto (plug-in) ognuno dei quali può essere considerato una «scatola nera».

Un programma così strutturato è facile da modificare in quanto ogni funzione è stata isolata. Le modifiche si riducono all'innesto di nuovi moduli; inoltre l'effetto di nuovi bachi rimane confinato in un ambito abbastanza ristretto. Si ricordi che la facilità di introdurre modifiche e la riduzione degli effetti di propagazione erano due dei compiti maggiori che ci si proponeva con la p.s.

Regole pratiche

Per finire si elencano delle regole di scomposizione a cui si è accennato prima, ma che ora appariranno più scontate.

Bisogna evitare nel limite del possibile dei moduli di inizializzazione: infatti questo toglie ai moduli inizializzati la caratteristica di «scatole nere».

Si esamini il progetto per trovare eventuali funzionioni duplicate; se, dopo averle eliminate, rimane del codice duplicato, questo è probabilmente necessario.

Si analizzino i moduli che sono chiamati o chiamano un numero molto elevato di altri moduli: probabilmente la scomposizione non è stata fatta correttamente.

I moduli che compiono funzioni simili probabilmente contengono delle sottofunzioni duplicate; se queste funzioni comuni possono essere isolate, le differenze possono spesso essere incorporate nelle routines chiamanti. Non esitate a scomporre ulteriormente un modulo: più tardi si scopre la necessità di spezzarlo, più difficile sarà poi interconnettere i vari pezzi.

Risultato finale

Ma è nella DOCUMENTAZIONE che la p.s. brilla particolarmente; se il progetto è stato fatto accuratamente non serve altra documentazione!

Perchè questo non sembri strano basta considerare ciò che il processo di raffinamento ha effettivamente prodotto: lo pseudocodice sostituisce il flow-chart, mentre il livello di specificazioni fornisce una descrizione tecnica completa. Introducendo, in modo opportuno, parte dello pseudocodice nel codice si ha un programma ben commentato. Forse null'altro se non una carta dei moduli e delle loro connessioni è necessario; ma questo è un prodotto gratuito della fase di scomposizione: finito il progetto è finita anche la documentazione.

Conclusione

La p.s. è un insieme di tecniche che aiuta ad organizzare lo sviluppo di programmi riducendoli ad una serie di passi facilmente gestibili. Il risultato finale è un programma documentato, ben fatto, facile da capire e manutenere. Incommensurabile è il guadagno in termini di fatica e tempo che può dare la PROGRAMMAZIONE STRUTTURATA. ■

Bibliografia

1) *Chip Weenos; «Designing Structured Programs» - Byte* n° 8-1978
2) Stevens, W P; Myers; G S; Constantine, LL; «Structured design» *IBM System Journal, 1974, numero 2, da pag. 115 a 135.*
3) *McClare; «Top-Down, Bottom-Up, and Structured Programming» IEEE Transaction on software engineering,* volume SE 1, numero 4 dicembre 1975, da pag. 397 a pag. 403
4) Wirth N: Principi di programmazione strutturata - *ISEDI*

Il fenomeno dei «Computer Shop»

di A. Cavalcoli

All'ultima BIAS abbiamo visto il nascere anche in Italia del fenomeno dei computer shop, su modello americano.

In realtà è sbagliato dire «su modello americano», e qui occorrerebbe una verifica a livello dibattito con i responsabili dei computer shop italiani.

Quindi, premesso che si tratta di una posizione personale, supportata da una serie di esperienze ed indagini, vediamo di esaminare le possibili implicazioni della presenza di queste iniziative in Italia.

Non si esclude certo che in Italia possa verificarsi lo stesso fenomeno ma stando alle esperienze di altri paesi europei, l'utenza è ancora lontana dall'essere costituita da una maggioranza di privati più o meno facoltosi e soprattutto vicini alla filosofia di impiego dei computer personali.

Alla computer boutique dell'avenue de Wagram, il negozio più grande di Parigi, l'80% dei clienti è costituito da piccole e medie industrie, con tutti i problemi che questo comporta.

Lo stesso dicasi per il computer shop Janal, sempre di Parigi, e per il computer workshop di Londra.

Lo stimolo maggiore che spinge il cliente verso il computer shop è la possibilità di scelta e di verifica personale dei vari sistemi, senza la necessità di più contatti polarizzati con le varie case costruttrici.

Vince quindi il supporto al cliente, supporto classico per la postvendita (garanzia, assistenza, laboratorio di assistenza), supporto dinamico e non sempre facile per la prevendita, in quanto il cliente non vuole solo che gli vengano illustrate le capacità operative dei sistemi, ma vuole provarli lui personalmente, magari sotto la guida di un esperto in time-sharing su più sistemi contemporaneamente.

A scoraggiare un dissennato approccio «time wasting» e come filtro atto ad eliminare i semplici curiosi, il computer workshop londinese permette un primo uso di assaggio pilotato dai sistemisti del negozio , dopodichè vengono vendute delle ore macchina, a prezzi ridotti, sia per chi vuole delle semplici esperienze, che per chi, ed allora si va ad appuntamento, vuole realizzare un intero progetto e fare dei conti a posteriori prima di impegnarsi nell'acquisto di più macchine.

E se il cliente chiede un'applicazione speciale, se vuole un software, se vuole interfacciarsi a qualche sistema già in suo possesso? Ecco saltare fuori dalla scrivania del negoziante una lunga lista di consulenti approvati, sia hardware che software: ed ancora una volta il supporto efficiente ha la meglio.

Evidentemente questa politica è già presente nella strategia degli shop italiani, la cui recente nascita non permette però un'analisi «di comportamento», come nel caso di altre iniziative, aventi alle spalle almeno uno o due anni di presenza sul mercato.

Cosa farà l'utente italiano? Sceglierà l'approccio da piccola-media industria, pilotato dagli esperti dei negozi?

Ancora, cosa faranno i negozianti di tutte le categorie? Getteranno nel retrobottega le calcolatrici e passeranno al sistema interattivo video-tastiera-floppy, con gestione di magazzino, contabilità generale e scadenziario crediti?

A Parigi, in occasione di una recente visita, ho assistito, nel negozio di amici importatori di articoli di regalo orientali, alla seguente scenetta:

Cliente: *«Avete ancora dei batik di Ceylon, quelli aventi come soggetto i danzatori?»*

Negoziante: *«Adesso vediamo» e rivolto al suo assistente «Guarda un po' sul computer cosa è rimasto»*

Breve sequenza sulla tastiera e sul display appare:

– Batik Ceylon:

- *Processione: 16 (3 grandi, 2 piccoli, 1 medio)*
- *Uccelli asiatici: 4*
- *Danzatori: in ordinazione (previsti 12/1/79)*
- *Ecc....*

Bene, ora si aspetta al varco il «ricchissimo salumiere» all'angolo di casa.

Giocare a Golf con il computer!

Nel numero precedente di BIT abbiamo riportato un programma BASIC completo di auto-apprendimento; un'altra possibile utilizzazione del personal computer è l'esecuzione di programmi che simulano giochi.

Possiamo avere ad esempio un programma per il gioco della dama: in questo caso uno dei giocatori siete voi ed il vostro avversario è il computer, oppure si possono avere giochi per una sola persona. In questo caso siete voi che giocate ed il computer vi fornisce informazioni sul risultato della vostra mossa.

Il programma che presentiamo qui di seguito fa parte di questa seconda categoria: simula una partita a golf.

È una partita a golf per un solo giocatore; in altre parole si tratta di combattere contro il campo di gioco (il calcolatore).

* Tratto da «Basic computer games» Creative Computing Press. Distribuito in Italia dalla Jackson Editrice s.r.l.

Il programma chiede qual'è il vostro handicap (il massimo è 30) e quali sono i vostri punti deboli.

Avete una collezione completa di 29 mazze, più un «putter» per i tiri da vicino.

Sul campo dovrete guardarvi da asperità del terreno, da alberi, da uscite di pista, da trappole di sabbia, da specchi d'acqua.

In più, può darsi che voi agganciate la palla, o che la colpiate di striscio, o che usciate dai limiti del campo, o che tiriate troppo lontano.

Nel green, determinerete l'efficacia del colpo in percentuale.

Finchè non conoscerete il campo di variabilità del gioco probabilmente sceglierete per l'handicap valori abbastanza alti.

Il programma è eseguibile usando il MITS Altair 8K BASIC, revisione 4.0.

GOLF

Esecuzione del programma

BENVENUTI AL GOLF CLUB CREATIVE COMPUTING,
UN CAMPO DI DICIOTTO BUCHE, DA CAMPIONATO,
SITUATO POCO LONTANO DA MORRISTOWN.
IL COMMENTATORE VI ASSISTE NEL GIOCO.
DIVERTITEVI!

QUAL'È IL VOSTRO HANDICAP? 10
I CATTIVI COLPI NEL GIOCO DEL GOLF COMPRENDONO:
0 = AGGANCIO DELLA PALLA,
1 = COLPO DI STRISCIO,
2 = TIRO CORTO,
4 = COLPI TRAPPOLA,
5 = TIRI ALL'INTERNO DEL GREEN.
QUAL'È IL VOSTRO (UNO SOLO) COLPO PEGGIORE? 1

SIETE ALL'INIZIO DELLA BUCA 1 PUNTEGGIO MEDIO 4
DISTANZA 361 METRI.
A DESTRA: PISTA VICINA
A SINISTRA: TERRENO SCONNESSO
SCELTA DELLA MAZZA

LUNGHEZZA DEL TIRO	MAZZE CONSIGLIATE
DA 200 A 280 METRI	DA 1 A 4
DA 100 A 200 METRI	DA 19 A 13
DA 0 A 100 METRI	DA 29 A 23

QUALE MAZZA VOLETE? 1

TIRO DI 237 METRI? CIOÈ 124 METRI DALLA BUCA.
LA PALLA È 10 METRI FUORI LINEA, IN PISTA
QUALE MAZZA VOLETE? 15

TROPPO FORTE. SIETE OLTRE LA BUCA.
TIRO DA 160 METRI, CIOÈ 36 METRI DALLA BUCA
LA PALLA È 0 METRI FUORI LINEA, IN PISTA
QUALE MAZZA VOLETE? 23

POTETE ORA MISURARE LA DISTANZA IN PERCENTUALE (DA 1 A 100).
PERCENTUALE DI VARIABILITÀ? 25

NEL GREEN, A 15 PALMI DALLA BUCA.
SCEGLIETE IL COEFFICIENTE DI DISTANZA PER IL PUTTER (DA 1 A 13).
COEFFICIENTE? 5
LA PALLA È PASSATA VICINO ALLA BUCA.
NEL GREEN, A 13 PALMI DALLA BUCA
SCEGLIETE IL COEFFICIENTE DI DISTANZA PER IL PUTTER (DA 1 A 13).

COEFFICIENTE? 3
PALLA IN BUCA.

IL PUNTEGGIO PER LA BUCA 1 È 5
IL PUNTEGGIO MEDIO TOTALE SU 1 BUCHE È 4, IL VOSTRO TOTALE È 5

SIETE ALL'INIZIO DELLA BUCA 2 PUNTEGGIO MEDIO 4
DISTANZA 389 METRI.
A DESTRA: ALBERI
A SINISTRA: ALBERI
QUALE MAZZA VOLETE? 2

TIRO DI 231 METRI, CIOÈ 158 METRI DALLA BUCA.
LA PALLA È 16 METRI FUORI LINEA, IN PISTA
QUALE MAZZA VOLETE? 14

NEL GREEN, A 18 PALMI DALLA BUCA.
SCEGLIETE IL COEFFICIENTE DI DISTANZA PER IL PUTTER (DA 1 A 13).
COEFFICIENTE? 7
LA PALLA È PASSATA VICINA ALLA BUCA.
NEL GREEN, A 17 PALMI DALLA BUCA.
SCEGLIETE IL COEFFICIENTE DI DISTANZA PER IL PUTTER (DA 1 A 13).
COEFFICIENTE? 5
LA PALLA È PASSATA VICINO ALLA BUCA

NEL GREEN, A 4 PALMI DALLA BUCA.
SCEGLIETE IL COEFFICIENTE DI DISTANZA PER IL PUTTER (DA 1 A 13).
COEFFICIENTE? 2
LA PALLA È PASSATA VICINO ALLA BUCA.
NEL GREEN, A 4 PALMI DALLA BUCA.
SCEGLIETE IL COEFFICIENTE DI DISTANZA PER IL PUTTER (DA 1 A 13).
COEFFICIENTE? 1
PALLA IN BUCA.

IL PUNTEGGIO PER LA BUCA 2 È 6
IL PUNTEGGIO MEDIO TOTALE SU 2 BUCHE È 8, IL VOSTRO TOTALE È 11

SIETE ALL'INIZIO DELLA BUCA 3 PUNTEGGIO MEDIO 3
DISTANZA 206 METRI.
A DESTRA: PISTA VICINA
A SINISTRA: TERRENO SCONNESSO
QUALE MAZZA VOLETE? 1

LA PALLA HA COLPITO UN ALBERO.
È RIMBALZATA SUL TERRENO SCONNESSO A 131 METRI DALLA BUCA.
QUALE MAZZA VOLETE? 16

AVETE COLPITO MALE LA PALLA.
TIRO DI 35 METRI, CIOÈ 96 METRI DALLA BUCA.
LA PALLA È 0 METRI FUORI LINEA, IN PISTA
QUALE MAZZA VOLETE? 23

POTETE ORA MISURARE LA DISTANZA IN PERCENTUALE (DA 1 A 100).
PERCENTUALE DI VARIABILITÀ? 75

TROPPO FORTE SIETE OLTRE LA BUCA.
TIRO DI 138 METRI, CIOÈ 43 METRI DALLA BUCA.
LA PALLA È 9 METRI FUORI LINEA, IN PISTA
QUALE MAZZA VOLETE? 24

POTETE ORA MISURARE LA DISTANZA IN PERCENTUALE (DA 1 A 100).
PERCENTUALE DI VARIABILITÀ? 30

Listing del programma

```
1    PRINT; PRINT; PRINT;
4    PRINT "BENVENUTI AL GOLF CLUB CREATIVE COMPUTING"
5    PRINT "UN CAMPO DI DICIOTTO BUCHE, DA CAMPIONATO,"
6    "SITUATO POCO LONTANO DA MORRISTOWN."
7    "IL COMMENTATORE VI ASSISTE NEL GIOCO."
8    "DIVERTITEVI!"
9    PRINT:PRINT:DIM L (10)
10   G1 =18
20   G2 =0
30   G3 =0
40   A=0
50   N =.8
60   S2 =0
70   F =1
80   PRINT "QUAL'È IL VOSTRO HANDICAP";
90   INPUT H
100  IF H >30 THEN 470
110  IF H <0 THEN 470
120  PRINT "I CATTIVI COLPI NEL GIOCO DEL GOLF COMPRENDONO
130  PRINT "0 =AGGANCIO DELLA PALLA,"
131  PRINT "1 =COLPO DI STRISCIO,"
132  PRINT "2 =TIRO CORTO,"
133  PRINT "4 =COLPI TRAPPOLA,"
134  PRINT "5 =TIRI ALL'INTERNO DEL GREEN."
140  PRINT "QUAL'È IL VOSTRO (UNO SOLO) COLPO PEGGIORE";
150  INPUT T
160  IF T >5 THEN 120
170  S1 =0
210  REM
230  L (0) =0
240  J =0
245  Q =0
250  S2 =S2 +1
260  K =0
270  IF F =1 THEN 315
290  PRINT "IL PUNTEGGIO PER LA BUCA"; F -1; "È"; S1
291  GOTO 1750
292  IF S1 =P THEN 299
293  IF S1 >P +2 THEN 297
294  IF S1 =P -1 THEN 301
295  IF S1 =P -2 THEN 303
296  GOTO 310
297  PRINT "SIETE UNA SCHIAPPA".
298  GOTO 310
299  PRINT "PUNTEGGIO UGUALE A QUELLO MEDIO. BENE."
300  GOTO 310
301  PRINT "SIETE MOLTO ABILE".
302  GOTO 310
303  IF P =3 THEN 306
304  PRINT "SIETE BRAVISSIMO".
305  GOTO 310
306  PRINT "BUCA IN UN TIRO".
310  IF F =19 THEN 1710
315  S1 =0
316  PRINT
320  IF S1 =0 THEN 1590
330  IF L (0) <1 THEN 1150
340  X =0
350  IF L (0) <5 THEN 1190
360  PRINT "TIRO DI;D1;"METRI, CIOÈ"; D2;"METRI DALLA BUCA."
362  PRINT "LA PALLA È"; INT (D); "METRI FUORI LINEA, IN";
380  GOSUB 400
390  GOTO 620
398  REM SUBROUTINE 400 PER DETERMINARE DOV'È LA PALLA,
399  REM O COSA C'È AI LATI DELLA PISTA.
400  IF L(X) =1 THEN 480
410  IF L (X) =2 THEN 500
420  IF L (X) =3 THEN 520
430  IF L(X) =4 THEN 540
440  IF L(X) =5 THEN 560
450  IF L(X) =6 THEN 580
460  PRINT "FUORI DAI LIMITI DEL CAMPO"
465  GOTO 1690
470  PRINT "HANDICAP: DA 0 A 30"
472  GOTO 8
480  PRINT "PISTA"
490  GOTO 1690
500  PRINT "TERRENO SCONNESSO"
510  GOTO 1690
520  PRINT "ALBERI"
530  GOTO 1690
540  PRINT "PISTA VICINA"
550  GOTO 1690
560  PRINT "TRAPPOLA"
570  GOTO 1690
580  PRINT "ACQUA"
590  GOTO 1690
591  REM FINE DELLA SUBROUTINE 400.
620  IF A =1 THEN 629
621  PRINT "SCELTA DELLA MAZZA"
622  PRINT "LUNGHEZZA DEL TIRO"; TAB (44);
     "MAZZE CONSIGLIATE"
623  PRINT "DA 200 A 280 METRI"; TAB (44); "DA 1 A 4"
624  PRINT "DA 100 A 200 METRI": TAB (44): "DA 19 A 13"
625  PRINT "DA 0 A 100 METRI";
     TAB (44); "DA 29 A 23"
626  A =1
629  PRINT "QUALE MAZZA VOLETE";
630  INPUT C
632  PRINT
635  IF C <1 THEN 690
637  IF C >29 THEN 690
640  IF C >4 THEN 710
650  IF L (0) <=5 THEN 740
660  IF C =14 THEN 740
665  IF C =23 THEN 740
670  GOTO 690
680  S1 =S1 -1
690  PRINT "LA MAZZA NON È DISPONIBILE".
693  PRINT
700  GOTO 620
710  IF C <12 THEN 690
720  C=C -6
730  GOTO 650
740  S1 =S1 +1
741  W =1
742  IF C >13 THEN 960
746  IF INT (F/3) =F/3 THEN 952
752  IF C<4 THEN 756
754  GOTO 760
756  IF L (0) =2 THEN 862
760  IF S1 >7 THEN 867
770  D1 =INT((30-H) *2.5 +187-((30-H)*.25+15)*C/
     2)+25*RND(1))
780  D1 =INT (D1 *W)
800  IF T =2 THEN 1170
830  Z =(RND (1)/.8)*(2 H+16) *ABS(TAN(D1*.0035))
840  D2 =INT (SQR(Z^2+ABS(D-D1)2))
850  IF D-D1 <0 THEN 870
860  GOTO 890
862  PRINT "AVETE COLPITO MALE LA PALLA."
864  D1 =35
866  GOTO 830
867  IF D <200 THEN 1300
868  GOTO 770
```

```
870   IF D2 <20 THEN 890
880   PRINT "TROPPO FORTE. SIETE OLTRE LA BUCA."
890   B = D
900   D = D2
910   IF D2 >27 THEN 1020
920   IF D2 >20 THEN 1100
930   IF D2 >.5 THEN 1120
940   L (0) =9
950   GOTO 1470
952   IF S2+Q+(10*(F—1)/18) <(F—1)*(72+((H+1)
      /.85))/18 THEN 956
954   GOTO 752
956   Q = Q + 1
957   IF S1/2 < >INT (S1 /2) THEN 1011
958   GOTO 862
960   PRINT "POTETE ORA MISURARE LA DISTANZA
      IN PERCENTUALE (DA 1 A 100)."
961   PRINT "PERCENTUALE DI VARIABILITÀ";
970   INPUT W
971   W = W /100
972   PRINT
980   IF W>1 THEN 680
985   IF L (0) =5 THEN 1280
990   IF C = 14 THEN 760
1000  C = C—10
1010  GOTO 760
1011  IF D <95 THEN 862
1012  PRINT "LA PALLA HA COLPITO UN ALBERO."
1013  PRINT "È RIMBALZATA SUL TERRENO
      SCONNESSO A"; D— 75;
1014  PRINT "METRI DALLA BUCA."
1015  D = D—75
1018  GOTO 620
1020  IF Z <30 THEN 1150
1022  IF J >0 THEN 1150
1030  IF T >0 THEN 1070
1035  S9 = (S2 + 1) /15
1036  IF INT (S9) =S9 THEN 1075
1040  PRINT "AVETE AGGANCIATO LA PALLA—";
1050  L(0) = L (2)
1055  IF Z >45 THEN 1092
1060  GOTO 320
1070  S9 = (S2 + 1) /15
1071  IF INT (S9) =S9 THEN 1040
1075  PRINT "AVETE COLPITO LA PALLA DI STRISCIO—";
1080  L(0) = L (1)
1090  GOTO 1055
1092  PRINT "MALE."
1094  GOTO 320
1100  L(0) =5
1110  GOTO 320
1120  L(0)=8
1130  D2 = INT (D2 *3)
1140  GOTO 1380
1150  L(0)=1
1160  GOTO 320
1170  D1 =INT.(.85 *D1)
1180  GOTO 830
1190  IF L (0) >6 THEN 1260
1200  PRINT "LA PALLA È CADUTA IN ACQUA."
1210  S1 =S1 +1
1220  PRINT "UN PUNTO DI PENALITÀ. RIPETETE IL TIRO."
1230  J = J + 1
1240  L(0) =1
1242  D = B
```

```
1250  GOTO 620
1260  PRINT "LA PALLA È USCITA DAL CAMPO".
1270  GOTO 1210
1280  IF T =3 THEN 1320
1300  D2 = 1 +(3*INT((80/(40—H))*RND(1)))
1310  GOTO 1380
1320  IF RND (1) >N THEN 1360
1330  N = N*. 2
1340  PRINT "COLPO MAL RIUSCITO,SIETE ANCORA IN TRAPPOLA."
1350  GOTO 620
1360  N =. 8
1370  GOTO 1300
1380  PRINT "NEL GREEN, A"; D2; "PALMI DALLA BUCA."
1381  PRINT SCEGLIETE IL COEFFICIENTE DI DISTANZA PER IL TIRO
1382  PRINT "PUTTER (DA 1 A 13)."
1383  PRINT "COEFFICIENTE";
1400  INPUT I
1410  S1 = S1 + 1
1420  IF S1 + 1—P > (H *.072) +2 THEN 1470
1425  IF K >2 THEN 1470
1428  K = K + 1
1430  IF.T = 4 THEN 1530
1440  D2 = D2 — I *(4 +2*RND(1)) +1.5
1450  IF D2<—2 THEN 1560
1460  IF D2>2 THEN 1500
1470  PRINT "PALLA IN BUCA."
1472  PRINT
1480  F = F + 1
1490  GOTO 230
1500  PRINT "TIRO CORTO."
1505  D2 = INT (D2)
1510  GOTO 1380
1530  D2 = D2—I *(4+1*RND(1)) +1
1550  GOTO 1450
1560  PRINT "LA PALLA È PASSATA VICINO ALLA BUCA."
1570  D2 = —D2
1580  GOTO 1505
1590  READ D,P,L (1), L(2)
1595  PRINT
1600  PRINT "SIETE ALL'INIZIO DELLA BUCA"; F;"
      PUNTEGGIO MEDIO"; P
1601  PRINT "DISTANZA"; D; "METRI."
1605  G3 = G3 + P
1620  PRINT "A DESTRA:";
1630  X =1
1640  GOSUB 400
1650  PRINT "A SINISTRA:";
1660  X = 2
1670  GOSUB 400
1680  GOTO 620
1690  RETURN
1699  REM
1700  DATA 361,4,4,2,389,4,4,3,206,3,4,2,500,5,7,2
1702  DATA 408,4,2,4,359,4,6,4,424,4,4,4,2,388,4,4,4
1704  DATA 196,3,7,2,400,4,7,2,560,5,7,2,132,3,2,2
1706  DATA 357,4,4,4,294,4,2,4,475,5,2,3,375,4,4,2
1708  DATA 180,3,6,2,550,5,6,6
1709  REM
1710  PRINT
1750  G2 = G2 + S1
1760  PRINT "IL PUNTEGGIO MEDIO TOTALE SU"; F—1; "BUCHE
      È"; G3:
1761  PRINT "IL VOSTRO TOTALE È"; G2
1762  IF G1 = F—1 THEN 1770
1765  GOTO 292
1770  END
```

- IL NASCOM 1 -

di **Ing. L. Grillo** – *Consulente Tecnico HOMIC s.r.l.*

«L'unico limite alle applicazioni dei microcomputer è la fantasia...» è lo slogan che si sente ripetere spesso. In questi ultimi anni abbiamo assistito ad una vera e propria corsa al miglioramento delle prestazioni e al ribasso dei prezzi dei microprocessori, ma purtroppo il costo delle periferiche non si è abbassato di molto.

Ad ogni modo i costruttori hanno cercato di minimizzare l'hardware più costoso riuscendo ad offrire microcomputers a prezzi molto accessibili senza sacrificarne le prestazioni.

Nei prodotti più economici, ad esempio, le periferiche fornite si riducono ad una tastiera numerica e ad un indicatore luminoso del tutto simili a quelli di una calcolatrice tascabile.

Ciò non riduce le capacità di elaborazione del microsistema ma complica le operazioni di compilazione e di prova dei programmi, richiedendo da parte del programmatore una certa concentrazione.

Nel valutare un microsistema non ci si potrà limitare quindi a verificarne la potenza di elaborazione (capacità di memoria, set di istruzioni, tempo di ciclo, ecc.) ma occorrerà valutare anche lo sforzo richiesto per compilare i programmi direttamente col microsistema. È infatti raro che un utente di home computer disponga di un sistema di sviluppo diverso.

Elenchiamo allora gli elementi che possono influire sulla scelta di un «home microcomputer system»:

- tipo di CPU usata (alla CPU è legata la disponibilità del software)
- interfacce standard disponibili nel sistema base
- sistema di sviluppo e software (loader, editor, debugger, compilatore assembler, interprete BASIC, utilities)
- espandibilità come memoria e come interfacce e, naturalmente, costo.

È difficile assegnare un peso ai vari elementi citati, e dare dei criteri di scelta. Piuttosto procediamo per esempi osservando come i vari costruttori hanno affrontato i problemi tecnico-commerciali.

La NASCO Ltd (London) si è prefissata di raggiungere i seguenti obiettivi:

- prezzo del microsistema sotto forma di kit di montaggio, inferiore al mezzo milione di lire
- facilità di utilizzo sia per applicazioni personali che per applicazioni industriali
- massima espandibilità sia come memoria che come periferiche
- massima flessibilità del software.

Il risultato è il NASCOM 1 costituito da un'unica piastra che contiene il potente micro Z80, la memoria, le interfacce delle periferiche ed una EPROM recante il monitor.

Il kit è completato da una tastiera alfanumerica già montata ed interfacciata. L'utente deve solo aggiungere l'alimentatore ed il proprio televisore. Un normale registratore a cassette serve a scrivere e a leggere i programmi ed i dati.

Altri elementi da valutare sono:

- Il micro Z80 è diffusissimo e sulle riviste specializzate si possono trovare programmi applicativi già pronti.
- L'hardware di base consente di lavorare senza costi aggiuntivi e con notevole elasticità. Ad esempio, oltre all'uscita UHF per TV domestico, è disponibile l'uscita video per monitor che dà una immagine ancora più chiara.
- La memoria a disposizione (fino a 64 K caratteri) consente di preparare ed eseguire programmi di notevole complessità.
- Comandi fissi permettono di controllare le elaborazioni attraverso messaggi molto chiari che compaiono sul monitor televisivo.
- Il software di base disponibile (assembler 3K, Tyni BASIC 2K, editor 1K) e le periferiche collegabili fanno del NASCOM 1 un vero e proprio sistema di sviluppo.

Diamo, per concludere, un elenco delle periferiche già interfacciate alla piastra NASCOM 1 base:

televisore domestico UHF 48 caratteri per 16 linee
MONITOR TV Video 48 caratteri per 16 linee
Tastiera alfanumerica 48 tasti a trasformatore d'impulso
Cassetta magnetica I/O 300 Bauds
Telescrivente Interfaccia seriale (loop di corrente) RS232
2 porte I/O parallele 16 Bit complessivi

Effettivamente, l'unico limite alle possibilità del microcomputer è la «fantasia».

■

Nuovo HP 250.

Fino ad ora, i computer a basso costo erano anche macchine abbastanza limitate, perché nessuno riusciva a rendere semplice un Sistema potente. Noi ci siamo riusciti: un vero elaboratore gestionale per 21 milioni!

Fino ad oggi gli elaboratori gestionali a basso costo erano anche caratterizzati da un basso livello di sofisticazione. Questo perché nessuno riusciva a far funzionare un Sistema complicato e potente in maniera semplice.

Noi ci siamo riusciti. L'HP 250 è incredibilmente semplice da usare, ma è anche veloce, ha sistema operativo e gestione della base di dati per eseguire operazioni complesse sia in linea che in batch: questo significa avere a disposizione un Sistema veloce e sofisticato senza bisogno di assumere un esperto per farlo funzionare.

Un nuovo elaboratore con gestione della base di dati e altre potenti funzioni software per l'ingresso e l'uscita dei dati, con memoria di Sistema fino a 192 Kbytes e memoria

La semplicità.

d'utente di 64 Kbytes, possibilità di auto-test, e prezzo che parte da 21 milioni.

Il nostro elegante computer è anche semplice da programmare: con il BASIC gestionale HP, l'IMAGE/250 (il software DBM), il QUERY/250, il FORMS/250 e il REPORT WRITER/250 hai a disposizione tutto per ottenere dal tuo Sistema il meglio in meno tempo.

Incorporati nello schermo regolabile ci sono otto tasti software estremamente utili: il significato di ognuno viene visualizzato sullo schermo, consentendo all'operatore di far girare un programma o di

inserirne uno nuovo in un istante.

Sia la memoria centrale che l'unità di elaborazione ad alta velocità sono sotto il piano della scrivania: e basta far scorrere lo sportello sul fianco per avere accesso all'unità a dischi flessibili (ne puoi usare fino a tre, o aggiungere dischi a cartuccia per aumentare la capacità di memorizzazione).

La tastiera dell'HP 250 è identica a quella di una macchina da scrivere, con la sezione numerica a 10 tasti uguale a quella delle addizionatrici: chiunque si sieda di fronte a questo elaboratore si sentirà immediatamente a suo agio.

Per i tabulati, puoi scegliere tra due modelli di stampanti.

E il modulo fonoassorbente opzionale è cosí effecace che puoi installare il Sistema perfino nell'ufficio del presidente.

Detto questo, devi convenire che il prezzo dell'HP 250 è una piacevole sorpresa.

Non vale la pena di telefonarci o di scriverci subito per saperne di piú?

Italia: Via Di Vittorio, 9 - 20063 Cernusco sul Naviglio (MI)
Tel. 903691 - Altri uff.: Roma, Torino, Padova, Bologna, Napoli

State pensando ad un microcomputer?

Assicuratevi di esaminare le offerte di prodotti della più grande Ditta del mondo, produttrice di una linea completa di Microelaboratori.
Tutte le macchine della Ohio Scientific sono dotate del più veloce e completo Basic-In-Rom o Su- Disco, ad impiego istantaneo, esistente nel campo della microelaborazione.

Serie Challenger I

Sistemi di elaborazione economici che parlano in Basic. Ideale per studenti, computer amatori, la scuola, la casa.
Superboard II - Il primo sistema al mondo completo, su una scheda comprendente tastiera, interfaccia per video display e audio-cassetta, BASIC-IN-ROM e fino a 8 K di memoria RAM.
Challenger IP - Superboard II completo di contenitore e con alimentatore incorporato.
Challenger IP - Disk - Sistema completo di mini-dischetti espandibile fino a 32 K di memoria RAM.

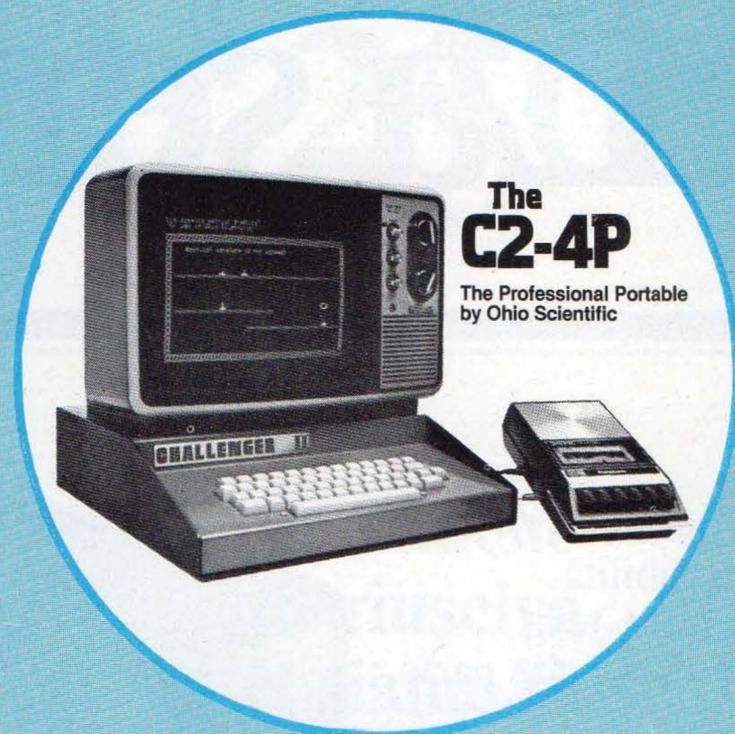
Serie Challenger II P

Microelaboratori, BUS orientati, ad altissime prestazioni, per impieghi personali, scolastici, di ricerca e nelle piccole aziende.
C2 - 4 P - Il portatile professionale
C2 - 8 P - La macchina personale più espandibile del mondo per impieghi commerciali e di ricerca.
C2 - 4 P - A mini - dischetti il meglio fra i portatili.
C2 - 8 P - A due dischetti per piccola azienda a un prezzo imbattibile.

Serie Challenger II con interfaccia seriale

Ha le stesse grandi caratteristiche della serie Challenger II P per coloro che hanno terminali seriali, utilizzata da piccole aziende, scuole, industrie.

Challenger III il meglio e il più potente nel campo dei microelaboratori

L'unico sistema a 3 processori (6502, 6800, Z80) per soddisfare le più sofisticate esigenze delle imprese
C3 - OEM - Il sistema a due dischetti compatto per grandi volumi di lavoro con 32 K di memoria RAM.
C3 - B - Il sistema basato sul disco Winchester da 74 M bytes. Il microelaboratore più potente del mondo.
La Ohio Scientific vi offre anche la più ampia gamma di accessori per l'espansione e la più grande scelta di software economico.
Confrontate un modello della Ohio Scientific con un modello comparabile di un'altra Casa che avete preso in considerazione. Confrontate le prestazioni, la reale possibilità di espansione, il software e i prezzi, e capirete perchè siamo diventati la più grande società di microelaboratori più estesa del mondo che produce la linea di sistemi.

PRODUTTORE

OHIO SCIENTIFIC

DISTRIBUTORE INTERNAZIONALE

AMERICAN DATA-HOME AND OFFICE COMPUTER

DISTRIBUTORE ESCLUSIVO PER L'ITALIA

EDICONSULT s.r.l.

OHIO SCIENTIFIC

DISTRIBUTORE

SKILAB

Sono interessato ai Microelaboratori OSI.
Inviatemi informazioni su:

- **Elaboratori personali**
- **Sistemi scolastici**
- **Elaboratori per piccole aziende**
- **Sistemi di sviluppo industriale**

Distributore esclusivo per l'Italia:

EDICONSULT s.r.l.

Via Caccini, 12 - 20052 MONZA - Tel. 039/389850

Nome..........
Indirizzo..........
Tel...........
Città.......... **CAP**..........

Il SORCERER della Exidy

Ing. Piero di Camillo *(Direttore Generale, Unicomp srl - Divisione Computeria)*

Il Sorcerer della Exidy Inc., è tra i personal computers di prezzo accessibile, quello che ha le caratteristiche tecniche più avanzate e le maggiori possibilità di espansione.

Nella versione base, il Sorcerer si presenta già con una memoria RAM di 8 K, un monitor di 4 K residente su ROM e uno Standard BASIC di 8 K, pure residente su ROM, che però ha il vantaggio di essere sostituibile con altri linguaggi, come vedremo in seguito. Sempre sulla versione base sono disponibili un'interfaccia seriale RS 232, un'interfaccia parallela, un'interfaccia per due registratori a cassette (del tipo normalmente usato per la registrazione e riproduzione di cassette audio) e un interfaccia per video con capacità di 1920 caratteri. Come video può essere utilizzato il televisore di casa, semplicemente attraverso un modulatore RF di basso costo.

Il computer Sorcerer si presenta con una tastiera compatta, all'interno della quale risiedono il microprocessore, la memoria e le interfacce già citate, e sul retro della quale sono presenti tutti i connettori e le boccole di ingresso uscita.

Figura 1 — Il Sorcerer (in italiano stregone) si presenta come una tastiera, entro la quale trovano posto il microprocessore Z 80, un monitor di 4 K su ROM, memoria RAM da 8K a 32K, un'interfaccia seriale e un'interfaccia parallela. Sul retro le boccole per il collegamento al video - che può essere il televisore di casa - e ad uno o due registratori a cassette. Sulla destra è visibile la cartuccia ROM PAC che contiene lo Standard Basic e che, se opportunamente sostituita, permette altri linguaggi di programmazione.

La tastiera presenta 128 caratteri ASCII e 128 simboli grafici, dei quali 64 già proposti dal costruttore e gli altri definibili dall'utente. In tal modo è possibile realizzare presentazioni grafiche anche complesse con risoluzione di 512 punti per 240, compresi alfabeti diversi e anche simboli di linguaggi speciali come l'APL.

Standard Basic e altri linguaggi

Lo Standard Basic viene fornito in una cartuccia brevettata, chiamata ROM PAC, che ha tutta l'aria di una cartuccia stereo a 8 tracce, ma che in realtà contiene una ROM. Tale cartuccia s'infila in una feritoia laterale del calcolatore e guadagna immediatamente il controllo del sistema all'atto dell'accensione.

Lo Standard Basic fornito dalla Exidy è un «interprete», il che vuol dire che non appena viene introdotto un programma, esso è pronto per l'elaborazione. Ciò è molto comodo per chi è alle prime armi, perchè si possono immediatamente identificare e correggere gli errori nei programmi, appena si caricano.

Lo Standard Basic è stato scritto per ottenere la massima efficienza, cioè il maggior numero di prestazioni con il minor ingombro di memoria. Si hanno così istruzioni multiple su una stessa riga, gestione completa delle stringhe e vettori a molte dimensioni, con una velocità di calcolo che permette circa 700 addizioni in virgola mobile al secondo, con una precisione di sette cifre.

In tabella 1 è riportato un elenco delle istruzioni dello Standard Basic. Da notare che è possibile chiamare delle routines scritte in linguaggio macchina, e che in qualsiasi momento l'utente può accedere al programma «monitor» per esaminare o modificare registri di memoria, stabilire punti di arresto sui programmi per la loro verifica ed eseguire un auto-collaudo delle condizioni di funzionamento.

Sostituendo la ROM PAC si può cambiare il linguaggio di programmazione. Ecco alcune delle ROM PAC che sono in via di sviluppo: Assembler, APL, Pilot, Word Processing. Il computer Sorcerer si può configurare anche come elaboratore dedicato, semplicemente caricando il software ad hoc realizzato in una cartuccia ROM PAC vergine.

Tabella 1 - Comandi e istruzioni dello Standard Basic

Istruzioni

Let	Rem	Fn	Wait
Then	Stop	Data	On..GoTo
GoTo	Def	Dim	If..GoTo
Restore	Print	GoSub	On..GoSub
Return	Next	Input	If..gosub
Read	Get	Out	
For	Step	End	

Comandi

Run	Null
List	CSave
New	CLoad
Clear	Peek
Cont	Poke

Funzioni matematiche

Abs	Int	Sgn	Tan
Atn	Inp	Sin	Usr
Cos	Log	Sqr	And
Exp	Pos	Spc	Or
Fre	Rnd	Tab	Not

Gestione delle stringhe

Asc	Str$
Len	Mid$
Val	Left$
Chr$	Right$

Segni di operazione (aritmetici e logici)

=	−	/\ <=>
+	*	↑<>>=<

Editing

, @	Control C	?
←	Control 0	: Return

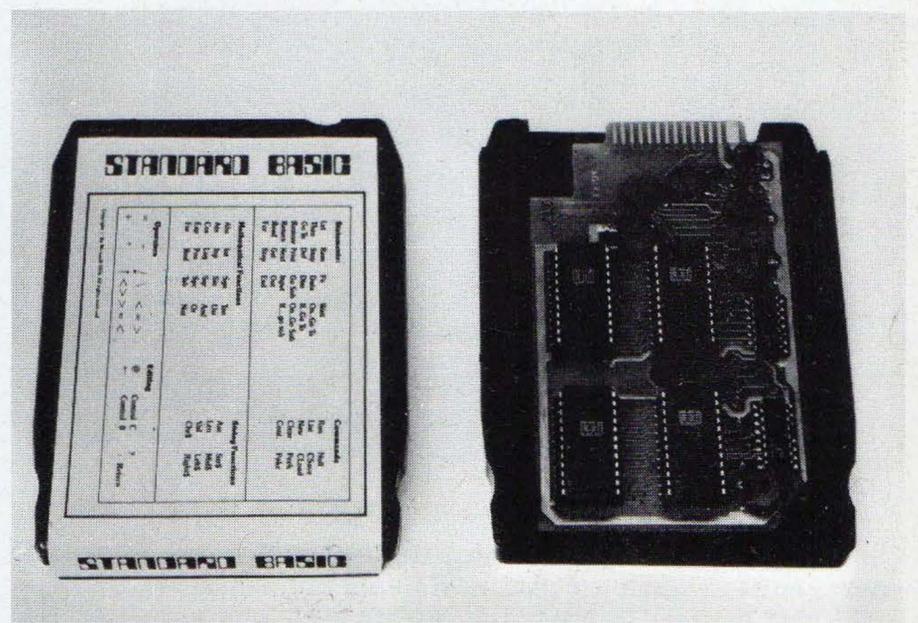



Figura 2 — La ROM-PAC brevettata che contiene su 8K lo Standard Basic. La cartuccia s'infila in una feritoia laterale del calcolatore, arricchendolo del linguaggio Basic.
Altre cartucce sono in via di sviluppo, per dotare il Sorcerer di altri linguaggi di programmazione.

Figura 3 — Il Sorcerer ha una tastiera professionale, del tipo di una macchina per scrivere, completa per la gestione delle informazioni, con una sezione di 16 tasti numerici e aritmetici (tipo calcolatrice) per velocizzare l'ingresso dei dati e le interrogazioni. La tastiera consiste di 79 tasti alfanumerici, con maiuscole e minuscole e caratteti grafici. Esistono tasti di «editing» e di controllo. È possibile assegnare qualsiasi simbolo o carattere speciale a qualsiasi tasto.

Tastiera e funzioni grafiche

Con il Sorcerer si possono facilmente realizzare rappresentazioni grafiche 512x240 per immagini ad elevata risoluzione. Ecco una nuova dimensione che si aggiunge alla programmazione: con i caratteri grafici del Sorcerer programmare delle immagini è facile come presentare messaggi sullo schermo. Anche un principiante, usando l'istruzione di stampa in BASIC, può costruire le immagini più fantasiose e fantastiche. La tastiera tipo macchina per scrivere ha un tasto per selezionare i simboli grafici incisi nella parte alta dei tasti, allo stesso modo che un tasto permette di selezionare le maiuscole in una normale macchina per scrivere. Si possono così usare i simboli grafici come fossero dei testi e creare l'immagine che si desidera presentare sullo schermo.

I simboli grafici definiti dall'utente permettono di esprimere la creatività del programmatore in simboli e forme geometriche, compresi alfabeti strani (cirillico ecc.) e perfino simboli per un nuovo linguaggio di programmazione. Sorcerer ha una selezione di 256 caratteri di espressione, dei quali 128 sono fissi e predefiniti, mentre gli altri 128 sono programmabili secondo la loro rappresentazione simbolica. Il carattere grafico è creato all'interno di una matrice di punti 8x8 e si può visualizzare sullo schermo qualsiasi combinazione dei 64, punti per carattere. 64 simboli grafici sono stati preprogrammati e rappresentati sulla tastiera per semplicità.

Gli altri 64 caratteri programmabili sono riservati alla creatività personale. Oppure anche tutti 128 i caratteri associati al tasto grafico possono essere definiti dall'utente. Con la grafica le qualità artistiche di ciascuno emergono e la grafica completamente definita dall'utente è quella che più permette una reale libertà di espressione.

Caratteri alfabetici (maiuscoli e minuscoli) e numerici sono inclusi nei 128 predefiniti e fissi di cui si è detto. La tastiera è così una vera e propria macchina per scrivere e si può usarla in modo conveniente per la scrittura di lettere o la correzione di testi. Con le sue trenta righe di testo, il Sorcerer presenta più informazioni sullo schermo di ogni altro elaboratore personale. Per essere precisi, in ogni istante si possono osservare 1920 caratteri prima di svolgere automaticamente la prossima pagina. Con la tastiera, comunicare con il Sorcerer è semplicissimo: introdurre informazioni o interrogare l'elaboratore è semplice come battere a macchina una lettera.

Caratteristiche tecniche

Modularità significa flessibilità. Il contenitore del Sorcerer, sul quale è ricavata la tastiera, contiene tutta l'elettronica che serve ad un sistema di elaborazione. Lo schema a blocchi in tabella 2 mostra le parti fornite come standard e quelle disponibili come opzioni.

Il microprocessore Z 80 è la CPU più avanzata disponibile su singolo chip. Esso è il cuore del Sorcerer ed è caratterizzato da un esteso set di istruzioni e una velocità di calcolo superiore a qualsiasi altro chip di CPU.

La memoria RAM è espandibile fino a 32 K. Anche il programmatore più sprecone non ha problemi. Il Sorcerer viene fornito normalmente con 8K di memoria RAM, completamente disponibile per i programmi di utente. Si possono acquistare memorie aggiuntive a blocchi di 16K per un totale di 32K. Si può così progredire tranquillamente in esperienza e sofisticazione: le maggiori esigenze di memoria di domani possono essere risolte dal Sorcerer.

Un normale registratore a cassette, trasferisce i dati sia a bassa (300 baud) che ad alta (1200 baud) velocità di trasmissione. Il controllo on/off del motore dell'unità a cassette è programmabile; sono collegabili fino a due unità.

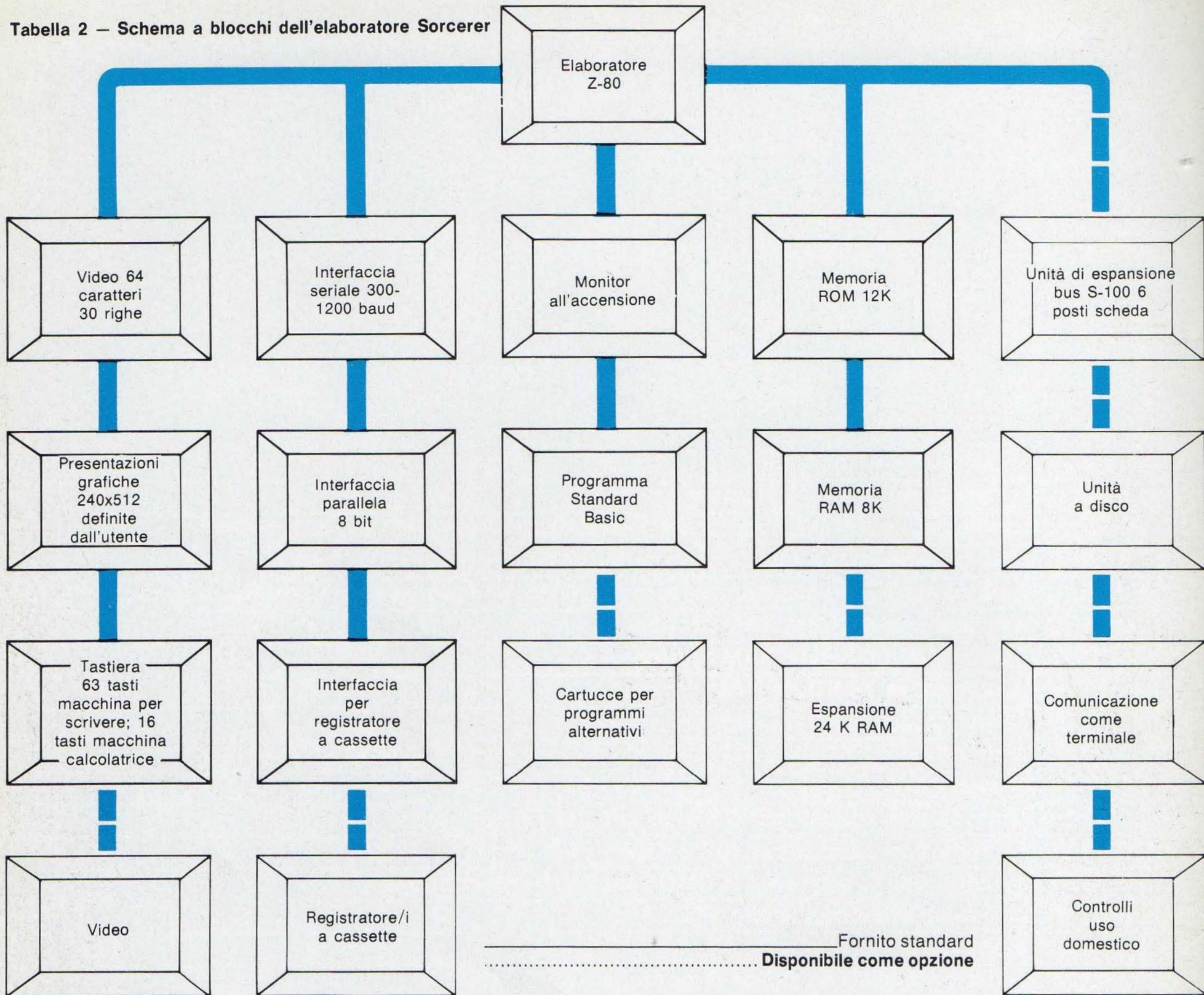
Un'interfaccia parallela consente il collegamento di qualsiasi unità periferica a 8 bit. Le linee di input/output sono bufferizzate e «latched» con controllo dello stato.

Un'interfaccia seriale fa del Sorcerer un terminale per altri elaboratori grandi o piccoli, professionali o personali, per via telefonica (Modem) o collegamento diretto.

Espandibilità e crescita sono innate nel Sorcerer. Prima o dopo può venire il momento in cui si desidera espandere il sistema. L'unità di espansione con il bus di I/O S-100 dà accesso a oltre 100 costruttori di dispositivi periferici. Si possono così collegare unità a dischi, schede di controllo industriale, dispositivi di comunicazione.

Espandibilità

Attraverso l'unità di espansione qualsiasi periferica S-100 compatibile può essere collegata al sistema Sorcerer che può crescere così fino a diventare un piccolo sistema gestionale di calcolo o di controllo di apparecchiature.

Tabella 2 — Schema a blocchi dell'elaboratore Sorcerer

Per espandere il Sorcerer con unità a floppy-disk può essere usato un sistema a floppy-disk o mini-floppy disk quali il North Star Micro-disk che fornisce un sistema completo basato su mini-floppy Shugart S 400 da 5" di diametro con il controller North Star. Questo sistema dà accesso a 90.000 caratteri formattati, gestiti da un completo software.

Tra le altre unità collegabili sono degne di attenzione i Metafloppy Micropolis che forniscono 2 minifloppy a quadrupla densità con capacità complessiva di 630 KByte formattati, con sistema operativo DOS e un BASIC molto potente.

Fra le unità S-100 compatibili collegabili a un sistema Sorcerer, particolarmente interessanti per impieghi hobbystici, professionali o di ricerca sono:

Software Technology Music System, un completo sistema hardware e software per la sintesi musicale. Si compone di un completo package software che occupa 2 KByte di una interfaccia S-100 compatibile che si inserisce nell'unità di espansione con un'uscita verso un amplificatore stereo e un altoparlante. Il sistema consente la sintesi di ottima musica attraverso un linguaggio di traduzione della notazione musicale.

Solid State Music Synthetizer Board è un sintetizzatore di forme d'onda che può essere interfacciato a un Bus S-100: esso è provvisto di un software MUS-X1 che consente la scrittura di musica polifonica con un linguaggio ad alto livello.

Mountain Hardware Introl è un sistema di controllo remoto di potenza. Con esso possono essere comandati dal Computer fino a 64 differenti apparecchi remoti senza che occorra aggiungere alcun altro cablaggio a quelli già esistenti in casa, in ufficio o nel laboratorio.

Heuristich Speechlab è un sistema a basso costo di riconoscimento della voce che consente input vocale su computer S-100 compatibili riconoscendo fino a 32 parole.

Computalker Speech Synthetizer esegue la sintesi da computer della voce.

Summagraphics Bit Pad è un digitizer a basso costo per il Sorcerer e per ogni computer S-100 compatibile.